ICAO

MACHINE READABLE TRAVEL DOCUMENTS TECHNICAL REPORT

RF PROTOCOL AND APPLICATION TEST STANDARD FOR EMRTD - PART 3

TESTS FOR APPLICATION PROTOCOL AND LOGICAL DATA STRUCTURE

Version – 3.00 | April 2023

ISO/IEC JTC 1/SC 17/WG 3/TF 4

FOR THE INTERNATIONAL CIVIL AVIATION ORGANIZATION

Release Control

Release	Date	Description
0.1	23-11-2005	First draft based on the German WG3 TF4 contribution "eMRTD Conformity Testing" version 1.02 presented at TF4 meeting in Paris Nov 21-23, 2005
0.2	21-12-2005	Updated version with new ICAO TR layout.
0.9	17-03-2006	Changes according to resolved comments from the WG3 TF4 meeting in Ottawa, Jan 30 – Feb 02, 2006. The following major changes have been introduced: • Less restrictive verification of status words • Introduction of profiles to be tested
0.95	2006-08-31	Intermediate draft, updated version with resolved comments from the Graz meeting Jun 12-13, 2006 and editorial changes. Some comments from Graz still unresolved. Test suites D and E have been modified as follows: There will be one test suite to check the protected command and one to check the unprotected commend. Test suites D and E will no longer check the correctness of the SM implementation since this is handled in test suite C New test suites F and G for testing unprotected SelectFile and ReadBinary. Test suites D and E will check the protected SelectFile and ReadBinary commands respectively. Redefined test cases (proposal) for test unit LDS_B – tests for DG1 – MRZ Redefined test cases E_1 – E_3 and E_5 – E_22 because of unspecified EOF reading.
0.96	2006-11-29	Final internal draft version including all resolved comments from the Bled meeting, Oct 25-26, 2006. Editorial changes in test suites C_9 and C_11 to clarify the encoding of offsets with tag 54. Editorial changes in test suites B and C. Former tests C_20 to C_36 have been moved to test suite B because these tests cover security condition tests.
1.0	2006-12-18	Editorial changes in 7816_C_8, C_9, and C_11 (sequence of steps to be performed) and in LDS_D_4 (reference changed to DOC9303, Annex A3.2)
1.01	2007-02-20	Test case 7816_C_16: verification of Postconditions removed
2.00 RC1	2013-02-14	Integration of contribution *AFNOR & BSI Contribution – SAC & AA conformity tests v1.6, January 13, 2013.*
2.00 RC2	2013-02-15	Update version of TR-03111
2.01	2013-02-28	ICAO submission
2.02 RC1	2013-08-02	Comments resolution
2.02 RC2	2013-09-05	Editorial modification in subclause 2.2
2.02 RC3	2013-10-01	Modification after comments resolution during Singapore TF4R meeting
2.03	2013-11-19	ICAO submission
2.04 RC1	2013-12-17	Comments resolution
2.04 RC2	2013-12-20	Review of comments resolution
2.05 RC1	2014-02-13	Comment on ISO7816_P_75 resolution
2.06	2014-03-10	ICAO submission
2.07	2014-10-10	Comments resolution after eMRTD Madrid event and Salamanca WG3
2.08 RC4	2016-05-26	Updates references to doc 9303 seventh edition PACE-CAM tests integration TF4R Berlin comments resolution Table 1 modification (remove line concerning 7816_P_82)
2.10	2016-07-07	ICAO submission
2.11	2018-03-23	Minor corrections Chip Authentication conformity tests addition Test Case 7816_R_07 addition on PACE-protected eMRTD 7816_P_18 removed (Reject unknown DO is not specified) Paris meeting 10-2017 WG3 Comments resolution IDEMIA comments resolution Veridos comments resolution TF4R japan meeting comments resolution
3.00	2023-04-05	Remove robustness tests Test unit ISO7816_A modification to cover PACE Only Chip Authentication tests are moved on ISO7816_I section instead of ISO7816_T. ISO7816_B_41, ISO7816_B_42: tests not applicable if EAC is supported Test unit ISO7816_F and ISO7816_G removal

		Test units ISO7816_H, ISO7816_J, ISO7816_K, ISO7816_L, ISO7816_M, ISO7816_N, ISO7816_T, ISO7816_U addition ISO7816_I_15, ISO7816_I_30 test cases addition: CA after PACE-CAM Test units LDS_F, LDS_G, LDS_H, LDS_L and LDS_M addition Test case LDS_B_04 modified to fit doc9303 part 4 only. Test case LDS_E_07 modified to remove SHA-1. Test case LDS_K_9 addition Test unit ISO7816_S modified: remove PACE-CAM OID from the test cases preconditions. Add ISO7816_S_5 to ISO7816_S_8 test cases. TF4R comments resolution (draft2) Remove Tag 7F2E in test cases LDS_C_09, LDS_G_10, LDS_H_10. **Note on this version:** **TA and CA under MF not covered** **Certificate sets for LDS2 not covered** **Migration to ISO/IEC 39794 not covered**

Table of contents

1 Introduction

1.1 Scope and purpose

An essential element of the ICAO compliant eMRTD is the addition of a Secure Contactless Integrated Circuit (SCIC) that holds securely biometric data of the eMRTD bearer within the ICAO defined Logical Data Structure (LDS).

Successful integration of the SCIC into the eMRTD depends upon active international cooperation between many companies and organizations.

The eMRTD has been specified and designed to operate correctly across a wide variety of reading infrastructures worldwide. The risk profile for the eMRTD indicates a high impact if that design includes a widespread error or fault. Therefore, it is essential, that all companies and organizations involved make all reasonable efforts to minimize the probability that this error or fault remains undetected before that design is approved and eMRTDs are issued.

This test specification covers the application interface, i.e. the ISO/IEC 7816 conformance of the eMRTD Chip and the conformance of the LDS.
The ISO/IEC 7816 conformance tests are restricted to the commands defined in the ICAO Doc 9303 document ([R1]). Other commands especially file creation and personalization commands are beyond the scope of this document.

The logical data structure test layer analyses the encoding of the LDS objects stored on an eMRTD. This layer contains several test units, one for each LDS1 eMRTD application's object (DG 1 - 16, EF.COM, EF.SOD, EF.CardAccess, EF.CardSecurity, EF.ATR/INFO and EF.DIR). Another test unit verifies the integrity and consistency of the different data structures.

Notes:

- The logical data structure of the LDS2 optional applications are not covered by this version of this document,
- the Terminal Authentication in the MF and Chip Authentication in the MF are not covered by this version of the document,
- This test specification addresses applicative layer functional aspects only. Security features are out of scope. The contactless interface tests are addressed by ISO/IEC18745-2.

1.2 Assumptions

It is assumed that the electrical interface and the underlying transport protocol are functionally tested. Thus, failures introduced by the RF protocol are out of scope of the test cases defined here.

1.3 Terminology

The key words "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [R2].

SHALL	This word, or the terms "REQUIRED" mean that the definition is an absolute requirement of the specification.
SHALL NOT	This phrase, mean that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even

useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY — This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option SHALL be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option SHALL be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1.4 Glossary

Command data field	The command data field defines the data of a command APDU that follows the command header and the Lc field – except the Le field. Its length is defined by Lc.
Command header	The command header comprises the first four bytes of the command APDU sent to the eMRTD in compliance with [R3]. The header consists of the bytes CLA, INS, P1, and P2.
Lc	Length field of the command APDU encoding the number of bytes in the command data field. In this specification, Lc is encoded in one byte (short length).
Le	Length field of the command APDU encoding the maximum number of bytes expected in the response data field. In this specification, Le is encoded in one byte (short length).
Response data	Response data is the string of bytes that is encoded in the response data field.
Response data field	The response data field defines the data – except the response trailer – that the eMRTD returns in the response APDU in compliance with [R3].
Response trailer	The response trailer defines the last two bytes that the eMRTD returns in the response APDU. The response trailer consists of two status bytes in compliance with [R3].
Status bytes	The status bytes SW indicate the processing state of the LDS application in compliance with [R3].
'80', 'AB CD'	Bytes or byte strings encoded in Hex-ASCII will be denoted in apostrophes.

1.5 Abbreviations

Abbreviation	
AA	Active authentication
AID	Application identifier
APDU	Application protocol data unit
AT	Authentication template
BAC	Basic access control
CA	Chip Authentication
CAM	Chip Authentication Mapping
CLA	Class byte
DF	Dedicated file
DG	Data group
DH	Diffie-Hellman
DO	Data object
EAC	Extended access control
ECDH	Elliptic Curve Diffie-Hellman
EF	Elementary file
FID	File identifier
ICS	Implementation conformance statement
INS	Instruction byte
KAEG	Key Agreement ElGamal-type
KAT	Key Agreement Template
LDS	Logical data structure
MRZ	Machine-readable zone
OID	Object identifier
P1, P2	Parameter bytes
PACE	Password Authenticated Connection Establishment referring Generic Mapping, Integrated Mapping and Chip Authentication Mapping
PCD	Proximity coupling device
PICC	Proximity integrated circuit card
PKD	Public-key directory
PKI	Public-key infrastructure
RF	Radio frequency
SCIC	Secure contactless integrated circuit
SFI	Short file identifier
SM	Secure Messaging
SOD	Security data object
SW	Status bytes
TA	Terminal Authentication
TBD	To be defined
TLV	Tag, length, value

1.6 Reference documentation

The following documentation served as reference for this technical report:

[R1] *ICAO Doc 9303 "Machine Readable Travel Documents" Eighth Edition — 2021*

[R2] *RFC 2119, S. Bradner, "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997*

[R3] *ISO/IEC 7816-4:2013. Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange.*

[R4] *ISO/IEC 19794-5:2005. Information technology -- Biometric data interchange formats -- Part 5: Face image data.*

[R5] *BSI TR-03111, Elliptic Curve Cryptography (ECC) Version 2.0, June 28, 2012*

[R6] *ISO/IEC 39794-5:2019. Information technology – Extensible biometric data interchange*

formats-- Part 5: Face image data.

[R7] *ISO/IEC 19794-4:2005. Information technology -- Biometric data interchange formats -- Part 4: Finger image data.*

[R8] *ISO/IEC 39794-4:2019. Information technology – Extensible biometric data interchange formats-- Part 4: Finger image data.*

[R9] *ISO/IEC 19794-6:2005. Information technology -- Biometric data interchange formats -- Part 6: IRIS image data.*

[R10] *ISO/IEC 39794-6:2021. Information technology – Extensible biometric data interchange formats-- Part 6: IRIS image data.*

2 General test requirements

The tests in this layer require a fully personalized eMRTD. This means that all mandatory LDS data groups SHALL be present.

This layer tests all mandatory ISO/IEC 7816 commands of the SCIC. There are additional test units testing optional features like BAC, PACE, CA, AA and TA.

All tests are mandatory unless marked as optional or conditional.

2.1 Test setup

2.1.1 General

For setting up these tests, any contactless reader supporting type A and type B protocols can be used. One personalized eMRTD sample is needed for executing the tests.
Most of the test cases in this document require an established PACE or BAC session and a selected LDS Application. In the preconditions of the test cases, this procedure is called “Open LDS Application”.

2.1.2 "Open LDS Application” procedure performed with BAC

The Open LDS Application procedure performed with BAC consists on following steps:

1. Reset the Chip
2. Select LDS Application
3. Perform BAC

2.1.3 "Open LDS Application” procedure performed with PACE

If the Open LDS Application procedure is performed with PACE, the MRZ SHALL be used. If PACE-CAM is supported, it SHALL NOT be used.
In this case, the Open LDS Application procedure performed with PACE consists on following steps:

1. Reset the Chip
2. Perform PACE (PACE-CAM SHALL NOT be used)
3. Select LDS Application

2.2 Implementation conformance statement

In order to set up the tests properly, an applicant SHALL provide the information specified in Table 1 below.

The ICAO specification defines several optional elements that an eMRTD can support. This includes security mechanisms like BAC, PACE, CA, AA and TA as well as additional LDS1 data groups (DG 3 to DG 16). Since these elements are optional, it is not possible to define the corresponding tests as mandatory for each eMRTD. Therefore, this document specifies a set of profiles. Each profile covers a specific optional element. A tested eMRTD SHALL be assigned to the supported profiles in the implementation conformance statement, and a test SHALL only be performed if the eMRTD belongs to this profile. The ICAO profile contains the mandatory feature set for ICAO compliant eMRTDs. Therefore, this profile and its tests are mandatory for all eMRTDs.

Note: There are no profile ID's explicitly defined for DG 14 and DG 15 because the PACE, CA, TA, AA profiles cover these LDS data groups implicitly as described below:

- DG15 is present in case of AA
- DG14 is present in case of PACE, CA, TA or AA-ECDSA. DG14 may also be present for additional mechanisms.

Table 1: ICS

Information for test setup	**Profile**	**Applicant declaration**
Access control applied:		

Information for test setup	Profile	Applicant declaration
• Basic Access Control • PACE	BAC PACE PACE-CAN PACE-DH PACE-ECDH PACE-IM PACE-GM PACE-CAM	
LDS1 Version	ICAO	
EF.DIR is present under MF	EFDIR	
Optional LDS2 applications are present	LDS2	
Read Binary with odd instruction byte supported	OddIns	
eMRTD contains elementary file with LDS1 Data Group 3	DG3	
eMRTD contains elementary file with LDS1 Data Group 4	DG4	
eMRTD contains elementary file with LDS1 Data Group 5	DG5	
eMRTD contains elementary file with LDS1 Data Group 6	DG6	
eMRTD contains elementary file with LDS1 Data Group 7	DG7	
eMRTD contains elementary file with LDS1 Data Group 8	DG8	
eMRTD contains elementary file with LDS1 Data Group 9	DG9	
eMRTD contains elementary file with LDS1 Data Group 10	DG10	
eMRTD contains elementary file with LDS1 Data Group 11	DG11	
eMRTD contains elementary file with LDS1 Data Group 12	DG12	
eMRTD contains elementary file with LDS1 Data Group 13	DG13	
eMRTD contains elementary file with LDS1 Data Group 16	DG16	
Authentication supported: • Passive Authentication • Active Authentication • Chip Authentication CA under LDS1 CA under MF CA with MSE:Set KAT CA with MSE:Set AT & General Authenticate CA based on Diffie-Hellman CA based on Elliptic Curve Diffie-Hellman CA may use Explicit key selection (KeyId is used in ChipAuthenticationInfo and ChipAuthenticationPublicKeyInfo) • Terminal Authentication TA under LDS1 TA under MF TA as used in European Union (use of EF.CVCA) Certificate date format validation supported	 ICAO AA AA-RSA AA-ECDSA CA CA-LDS1 CA-MF CA-KAT CA-ATGA CA-DH CA-ECDH CA-KEYREF TA TA-LDS1 TA-MF TA-EU TA-DATE	
MRZ provided with the samples	ICAO	
Country signing certificate used to verify EF.SOD and EF.CardSecurity (if applicable)	ICAO	
Configuration list described in the EF.CardAccess	PACE	*(Algorithm OID + Domain parameters)*
Invalid key reference for PACE (used in test case ISO7816_P_09)	PACE	
Invalid password identifier for PACE (used in test case ISO7816_P_08)	PACE	
Valid PACE OID not supported by the eMRTD (used in test case ISO7816_P_68. If such an OID does not exist, ISO7816_P_68 is not applicable)	PACE	

Information for test setup	Profile	Applicant declaration
Configuration list described in the EF.DG14 (required and conditional SecurityInfos)	CA	*(Algorithm OID + Public Key parameters)*
Command to send to the eMRTD to verify the chip's ability to still require Secured APDU. If not provided, use '00 B0 81 00 00'.	PACE CA	

The test cases reference the profiles which define a precondition for the test execution.

2.3 Verification of ISO/IEC 7816-4 status bytes

For most of test cases defined in this document, the status bytes returned by the eMRTD are not exactly defined in the ICAO specification. In these cases, the result analysis uses the scheme defined in [R3] to specify the expected result. It is only checked that the response belongs to the specified category. In cases where the expected result is unambiguously defined in the ICAO specification, the exact value is specified in the test case.

Proprietary status bytes outside the range of defined ISO status bytes will be treated as failures in the test cases.

Table 2: ISO/IEC 7816-4 status bytes

Status bytes category	Category name	Valid value range	Process behavior
Normal processing	Normal processing status bytes	'90 00' '61 XX'	Process completed
Warning processing	ISO warning	'62 XX' '63 XX'	Process completed
Execution error	ISO execution error	'64 XX' '65 XX' '66 XX'	Process aborted
Checking error	ISO checking error	'67 XX' '68 XX' '69 XX' '6A XX' '6B XX' '6C XX' '6D XX' '6E XX' '6F XX'	Process aborted

Note: There is a significant difference between normal and warning processing on the one side and execution and checking error on the other side. The first group is returned if the process has been fully completed, and the eMRTD MAY return some additional data. The “process aborted” categories are issued if the command cannot be performed. Therefore, response data SHALL NOT be returned. In all test cases where an execution or checking error is expected, it SHALL be verified that the eMRTD does not return any response data except SM protocol elements (DO '99' / '8E').

2.4 Key pair definition

The certificate sets defined in chapter 2.5 are based on several asymmetric key pairs. In preparation to the tests, these key pairs have to be generated. The parameter used for these keys are depending on the initial CVCA private key.
The initial CVCA root private key SHOULD be provided by the eMRTD vendor. It is also possible the vendor generates all keys and certificates on its own and passes it to the test operator for the tests.
All key pairs SHALL be generated independently, so it is not permitted to use the same key pair for all sets.

Key pair	Definition

CVCA_KEY_00	The key pair CV_KEY_00 is the public/private key for the initial CVCA root.
CVCA_KEY_05_1	The key pair CVCA_KEY_05_1 is the public/private key for the CVCA_05_1.
CVCA_KEY_05_2	The key pair CVCA_KEY_05_2 is the public/private key for the CVCA_05_2.
CVCA_KEY_06_1	The key pair CVCA_KEY_06_1 is the public/private key for the CVCA_06_1.
CVCA_KEY_06_2	The key pair CVCA_KEY_06_2 is the public/private key for the CVCA_06_2.
CVCA_KEY_07	The key pair CVCA_KEY_07 is the public/private key for the CVCA_07.
DV_KEY_01	Key pair of the test certificates DV_CERT_1_x
DV_KEY_01_1	Key pair of test certificate DV_CERT_1_32 (SHALL be shorter than CVCA Key length)
DV_KEY_02	Key pair of the test certificates DV_CERT_2_x
DV_KEY_03	Key pair of the test certificates DV_CERT_3_x
DV_KEY_04	Key pair of the test certificates DV_CERT_4_x
DV_KEY_05	Key pair of the test certificates DV_CERT_5
IS_KEY_01	Key pair of the test certificates T_CERT_1_x
IS_KEY_01_1	Key pair of test certificate T_CERT_1_6 (SHALL be shorter than CVCA Key length)
IS_KEY_02	Key pair of the test certificates T_CERT_2_x except T_CERT_2_21_1
IS_KEY_02_1	Key pair of the test certificates T_CERT_2_21_1
IS_KEY_03	Key pair of the test certificates T_CERT_3_x
IS_KEY_04	Key pair of the test certificates T_CERT_4_x
IS_KEY_05	Key pair of the test certificates T_CERT_5

Table 1 – Key pair definition

2.5 Certificate specification

2.5.1 General

Since the advanced security mechanisms are using a certificate-based authentication schema it is necessary to provide a set of well-prepared certificates in order to perform all tests.
This chapter defines the exact set of certificates referred in the tests. Besides the regular certificate chain there is also the need for special encoded certificates.
The certificates are specified in two different ways. For provider of personalized eMRTD samples, which do already have a preconfigured trust point based on their own CVCA key pair, the chapters below define a set of certificates relative to the effective date ($CVCA_{eff}$) and expiration date ($CVCA_{exp}$) of the given the CVCA. The time span between $CVCA_{eff}$ and $CVCA_{exp}$ SHALL be at least two months to allow proper adoption of the certificate time scheme defined below. The "current date" of the provided sample SHALL be set to $CVCA_{eff}$ before the tests are started. The provider of the sample or the test laboratory has to generate the corresponding certificate according to this specification based on the CVCA data.
If no preconfigured key pair is available or if the production process allows the use of an externally defined CVCA, a certificate set can be used which is defined as a "worked example" by this specification. This set is provided for ECDSA, RSA and RSAPSS based certificates and is defined in a full binary form with fixed keys and dates. It also includes a definition for an initial CVCA key pair and its effective and expiry dates.

2.5.2 Certificate Set 1

The certificate set consist of a regular certificate chain (DV -> IS) for LDS1 Application which is used for the positive tests regarding the certificate verification and Terminal Authentication. Furthermore, it contains variants of the original DV certificate to simulate a variety of certificate coding issues (missing elements, badly encoded dates …).

2.5.2.1 LINK_CERT_1_1

Purpose	This certificate is an irregular CVCA certificate. The signing key is a DV key

Version	1.0	
Referred by	ISO7816_J_39, ISO7816_J_40	
Parameter	Certificate Authority Reference	DETESTDVDE001
	Certificate Holder Reference	As defined by the CVCA root
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{exp}$
	Public Key reference	Public key of key pair CVCA _KEY_00
	Signing Key reference	Signed with the private key of key pair DV_KEY_01

2.5.2.2 LINK_CERT_1_2

Purpose	This certificate is an irregular CVCA certificate. The signing key is an IS key	
Version	1.0	
Referred by	ISO7816_J_48	
Parameter	Certificate Authority Reference	DETESTTDE001
	Certificate Holder Reference	As defined by the CVCA root
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{exp}$
	Public Key reference	Public key of key pair CVCA _KEY_00
	Signing Key reference	Signed with the private key of key pair IS_KEY_01

2.5.2.3 DV_CERT_1

Purpose	This certificate is a regular DV certificate for ePassport Application, which validity period starts at the effective date of the CVCA and expires after one month.	
Version	1.0	
Referred by	ISO7816_J_1, ISO7816_J_2, ISO7816_J_3, ISO7816_J_4, ISO7816_J_5, ISO7816_J_12, ISO7816_J_13, ISO7816_J_14, ISO7816_J_18, ISO7816_J_21, ISO7816_J_22, ISO7816_J_23, ISO7816_J_24, ISO7816_J_25, ISO7816_J_26, ISO7816_J_27, ISO7816_J_28, ISO7816_J_29, ISO7816_J_30, ISO7816_J_31, ISO7816_J_32, ISO7816_J_33, ISO7816_J_34, ISO7816_J_35, ISO7816_J_36, ISO7816_J_37, ISO7816_J_39, ISO7816_J_41, ISO7816_J_42, ISO7816_J_45, ISO7816_J_46, ISO7816_J_47, ISO7816_J_48, ISO7816_J_50, ISO7816_K_1, ISO7816_K_2, ISO7816_K_3, ISO7816_K_4, ISO7816_K_7, ISO7816_K_8, ISO7816_K_9, ISO7816_K_10, ISO7816_K_11, ISO7816_K_12, ISO7816_K_13, ISO7816_K_14 ISO7816_L_9, ISO7816_L_10, ISO7816_L_11, ISO7816_L_12, ISO7816_L_13, ISO7816_L_14	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG3, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_01
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.2.4 DV_CERT_1_1

Purpose	This certificate is similar to DV_CERT_1, does not contain a Certificate Holder Authorization	
Version	1.0	
Referred by	ISO7816_J_6	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	**Absent**
	Certificate effective date	See DV_CERT_1

	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.5 DV_CERT_1_2

Purpose	This certificate is similar to DV_CERT_1, but does not contain a Certificate Effective Date	
Version	1.0	
Referred by	ISO7816_J_7	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	**Absent**
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.6 DV_CERT_1_3

Purpose	This certificate is similar to DV_CERT_1, but does not contain a Certificate Expiration Date	
Version	1.0	
Referred by	ISO7816_J_8	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	**Absent**
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.7 DV_CERT_1_4

Purpose	This certificate is similar to DV_CERT_1, but contain a badly encoded Certificate Effective Date (Invalid BDC encoding)	
Version	1.0	
Referred by	ISO7816_J_9	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	**0A 0B 0C 0D 0E 0F (invalid BCD encoding)**
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.8 DV_CERT_1_5

Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Expiration Date (invalid BCD encoding)	
Version	1.0	
Referred by	ISO7816_J_10	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	**0A 0B 0C 0D 0E 0F (invalid BCD encoding)**
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.9 DV_CERT_1_6

Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Effective Date (Invalid Gregorian date)
Version	1.0

Referred by	ISO7816_J_15	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	**The month and the year used as defined by the $CVCA_{eff}$ and the day is always set to the 32nd so that it becomes an invalid Gregorian date.**
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.10 DV_CERT_1_7

Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Expiration Date (Invalid Gregorian date)	
Version	1.0	
Referred by	ISO7816_J_16	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	**The month and the year used as defined by the $CVCA_{eff}$ and the day is always set to the 32nd so that it becomes an invalid Gregorian date.**
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.11 DV_CERT_1_8

Purpose	This certificate is similar to DV_CERT_1, but contains a Certificate Expiration Date BEFORE the Certificate Effective Date	
Version	1.0	
Referred by	ISO7816_J_17	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	**$CVCA_{eff}$ + 1 day**
	Certificate expiration date	**$CVCA_{eff}$**
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.12 DV_CERT_1_9

Purpose	This certificate is similar to DV_CERT_1, but contains a Certificate Holder Authorization with an invalid OID	
Version	1.0	
Referred by	ISO7816_J_19	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1 Use OID 04 00 7F 00 07 03 01 02
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.13 DV_CERT_1_10

Purpose	This certificate is similar to DV_CERT_1, but contains a Public Key with an invalid OID
Version	1.0

Referred by	ISO7816_J_20	
Content definition	See DV_CERT_1 **The Certificate Public Key is used with an invalid OID (Use 0.4.0.127.0.7.2.2.2.5.1)**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.14 DV_CERT_1_11

Purpose	This certificate is similar to DV_CERT_1, but it's a DV foreign certificate	
Version	1.0	
Referred by	ISO7816_J_11, ISO7816_J_40, ISO7816_J_43, ISO7816_J_44	
Content definition	See DV_CERT_1 **Role is set to foreign DV**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	Foreign DV
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.15 DV_CERT_1_12

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days.	
Version	1.0	
Referred by	ISO7816_J_21, ISO7816_J_31	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.16 DV_CERT_1_13

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a wrong "certificate body" tag	
Version	1.0	
Referred by	ISO7816_J_22	
Content definition	See DV_CERT_1 Use 7F 4F instaed of 7F 4E	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.17 DV_CERT_1_14

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a wrong "certificate signature" tag	
Version	1.0	

Referred by	ISO7816_J_23	
Content definition	See DV_CERT_1 Use 5F 38 instead of 5F 37	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.18 DV_CERT_1_15

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with an inconsistent "certificate body" D.O. (wrong length)	
Version	1.0	
Referred by	ISO7816_J_24	
Content definition	See DV_CERT_1 *bb* is the encoded length of the certificate body object decreased by one	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.19 DV_CERT_1_16

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with an inconsistent "certificate signature" D.O. (wrong length)	
Version	1.0	
Referred by	ISO7816_J_25	
Content definition	See DV_CERT_1 *ii* is the encoded length of the certificate signature object decreased by one *jj* is the placeholder for the certificates signature (ii **+ 1** bytes)	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.20 DV_CERT_1_17

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with a wrong signature	
Version	1.0	
Referred by	ISO7816_J_26	
Content definition	See DV_CERT_1 *jj* is the placeholder for the certificates signature (ii bytes) **last byte is increased by one (mod 256)**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.21 DV_CERT_1_18

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with a wrong signature	
Version	1.0	
Referred by	ISO7816_J_27	
Content definition	See DV_CERT_1 *jj* is the placeholder for the certificates signature (ii bytes) **last byte is dropped and *ii* is updated according to the new length**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.22 DV_CERT_1_19

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with a wrong signature (Profile RSA only)	
Version	1.0	
Referred by	ISO7816_J_28	
Content definition	See DV_CERT_1 *jj* is the placeholder for the certificates signature (ii bytes) **the signature is greater than the modulus of the issuing key CVCA_KEY_00, the length of signature matches the length of the modulus**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.23 DV_CERT_1_20

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with a wrong signature (Profile ECDSA only) The wrong signature is obtained by filling the 'r' part of the signature with '00'. The length of 'r' is still matches the size of the prime.	
Version	1.0	
Referred by	ISO7816_J_29	
Content definition	See DV_CERT_1 *jj* is the placeholder for the certificates signature (ii bytes) **with 'r' = 0**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.24 DV_CERT_1_21

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate with a wrong signature (Profile ECDSA only) The wrong signature is obtained by filling the 's' part of the signature with '00'. The length of 's' is still matches the size of the prime.	
Version	1.0	
Referred by	ISO7816_J_30	
Content definition	See DV_CERT_1 *jj* is the placeholder for the certificates signature (ii bytes) **with 's' = 0**	
Parameter	Certificate Authority Reference	See DV_CERT_1

	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.25 DV_CERT_1_22

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate Public Key with a wrong O.I.D.	
Version	1.0	
Referred by	ISO7816_J_32	
Content definition	See DV_CERT_1 *ff* is the placeholder for the certificates public key bytes (ee bytes) **THE O.I.D. has** an **uncorrect value that does not indicate id-TA : (0.4.0.127.0.7.2.2.3.x.y)**,	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.26 DV_CERT_1_23

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate Public Key with a missing O.I.D.	
Version	1.0	
Referred by	ISO7816_J_33	
Content definition	See DV_CERT_1 *ff* is the placeholder for the certificates public key bytes (ee bytes), **It does not contain any O.I.D Data Object**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.27 DV_CERT_1_24

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate Public Key with a missing elliptic curve public point (Profile ECDSA only)	
Version	1.0	
Referred by	ISO7816_J_34	
Content definition	See DV_CERT_1 *ff* is the placeholder for the certificates public key bytes (ee bytes), **The elliptic curve public point is missing**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.28 DV_CERT_1_25

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days.

	It contains a Certificate Public Key with a missing RSA modulus (Profile RSA only)	
Version	1.0	
Referred by	ISO7816_J_35	
Content definition	See DV_CERT_1 *ff* is the placeholder for the certificates public key bytes (ee bytes), **The RSA modulus is missing**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.29 DV_CERT_1_26

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate Public Key with a wrong missing RSA public exponent (Profile RSA only)	
Version	1.0	
Referred by	ISO7816_J_36	
Content definition	See DV_CERT_1 *ff* is the placeholder for the certificates public key bytes (ee bytes), **The RSA public exponent is missing**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.30 DV_CERT_1_27

Purpose	This certificate is similar to DV_CERT_1, but validity period starts at the effective date of the CVCA added to one month and twenty days, expiration date is fixed after 5 days. It contains a Certificate Public Key with an unknown D.O Profile ECDSA within EC parameters (tag '77') Profile RSA within RSA parameters ('77 01 00')	
Version	1.0	
Referred by	ISO7816_J_37	
Content definition	See DV_CERT_1 *ff* is the placeholder for the certificates public key bytes (ee bytes), **An unknown Data Object**	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	See DV_CERT_1
	Signing Key reference	See DV_CERT_1

2.5.2.31 DV_CERT_1_28

Purpose	This certificate is similar to DV_CERT_1, but it's an irregular DV domestic certificate. The signing key is a DV key.	
Version	1.0	
Referred by	ISO7816_J_41, ISO7816_J_43	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	DETESTDVDE001
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	Signed with the private key of key pair DV_KEY_01

2.5.2.32 DV_CERT_1_29

Purpose	This certificate is similar to DV_CERT_1_11, but it's an irregular foreign DV certificate. The signing key is a DV key.	
Version	1.0	
Referred by	ISO7816_J_42, ISO7816_J_44	
Content definition	See DV_CERT_1_11	
Parameter	Certificate Authority Reference	DETESTDVDE001
	Certificate Holder Reference	See DV_CERT_1_11
	Certificate Holder Authorization	See DV_CERT_1_11
	Certificate effective date	See DV_CERT_1_11
	Certificate expiration date	See DV_CERT_1_11
	Public Key reference	See DV_CERT_1_11
	Signing Key reference	Signed with the private key of key pair DV_KEY_01

2.5.2.33 DV_CERT_1_30

Purpose	This certificate is similar to DV_CERT_1_11, but it's an irregular foreign DV certificate. The signing key is an IS key.	
Version	1.0	
Referred by	ISO7816_J_45	
Content definition	See DV_CERT_1_11	
Parameter	Certificate Authority Reference	DETESTTDE001
	Certificate Holder Reference	See DV_CERT_1_11
	Certificate Holder Authorization	See DV_CERT_1_11
	Certificate effective date	See DV_CERT_1_11
	Certificate expiration date	See DV_CERT_1_11
	Public Key reference	See DV_CERT_1_11
	Signing Key reference	Signed with the private key of key pair IS_KEY_01

2.5.2.34 DV_CERT_1_31

Purpose	This certificate is similar to DV_CERT_1, but it's an irregular domestic DV certificate. The signing key is an IS key.	
Version	1.0	
Referred by	ISO7816_J_46	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	DETESTTDE001
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1
	Certificate expiration date	See DV_CERT_1
	Public Key reference	See DV_CERT_1
	Signing Key reference	Signed with the private key of key pair IS_KEY_01

2.5.2.35 DV_CERT_1_32

Purpose	This certificate is similar to DV_CERT_1, but contains a Certificate Public Key with a short key. For RSA profile, same Algorithm Identifier but PK.DVCA's modulus length is shorter than the CVCA's key modulus length. For ECDSA profile, same Algorithm Identifier but DVCA's domain parameters are different and have a shorter prime length than the CVCA's key. The hash algorithm should be adapted if necessary.	
Version	1.0	
Referred by	ISO7816_J_49	
Content definition	See DV_CERT_1	
Parameter	Certificate Authority Reference	See DV_CERT_1
	Certificate Holder Reference	See DV_CERT_1
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	See DV_CERT_1

	Certificate expiration date	See DV_CERT_1
	Public Key reference	Public key of key pair DV_KEY_01_1 (shorter than DV_KEY_01)
	Signing Key reference	See DV_CERT_1

2.5.2.36 T_CERT_1

Purpose	This certificate is a regular IS certificate for ePassport Application, which validity period starts at the effective date of the CVCA and expires after 14 days.	
Version	1.0	
Referred by	ISO7816_J_1, ISO7816_J_2, ISO7816_J_3, ISO7816_J_4, ISO7816_J_5, ISO7816_J_6, ISO7816_J_7, ISO7816_J_8, ISO7816_J_9, ISO7816_J_10, ISO7816_J_13, ISO7816_J_14, ISO7816_J_15, ISO7816_J_16, ISO7816_J_17, ISO7816_J_18, ISO7816_J_19, ISO7816_J_20 ISO7816_J_45, ISO7816_J_46, ISO7816_J_47, ISO7816_J_48, ISO7816_K_1, ISO7816_K_2, ISO7816_K_3, ISO7816_K_7, ISO7816_K_8, ISO7816_K_9, ISO7816_K_10, ISO7816_K_11, ISO7816_K_12, ISO7816_K_13, ISO7816_K_14 ISO7816_L_9, ISO7816_L_10, ISO7816_L_11, ISO7816_L_12, ISO7816_L_13, ISO7816_L_14	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** 0D 44 45 54 45 53 54 44 56 44 45 30 30 31 **7F 49** *ee ff* **5F 20** 0D 44 45 54 45 53 54 54 44 45 30 30 31 **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 03 **5F 25** 06 *gg* **5F 24** 06 *hh* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length the certificate body object *ee* is the encoded length of the certificates public key, *ff* is the placeholder for the certificates public key bytes (ee bytes), *gg* is the placeholder for the BCD encoded effective date of the certificate *hh* is the placeholder for the BCD encoded expiration date of the certificate *ii* is the encoded length of the certificates signature object, *jj* is the placeholder for the certificates signature (ii bytes)	
Parameter	Certificate Authority Reference	DETESTDVDE001
	Certificate Holder Reference	DETESTTDE001
	Certificate Holder Authorization	ePassport Application Inspection system, DG3, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 14 days
	Public Key reference	Public key of key pair IS_KEY_01
	Signing Key reference	Signed with the private key of key pair DV_KEY_01

2.5.2.37 T_CERT_1_1

Purpose	This certificate is a regular Terminal certificate, which is issued by the DV_CERT_1_10. It has an advanced effective date. (Beyond the expiration date of T_CERT_1_2)	
Version	1.0	
Referred by	ISO7816_J_11	
Content definition	See T_CERT_1	
Parameter	Certificate Authority Reference	See T_CERT_1
	Certificate Holder Reference	See T_CERT_1
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 14 days

	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	See T_CERT_1
	Signing Key reference	See T_CERT_1

2.5.2.38 T_CERT_1_2

Purpose	This certificate is a regular AT certificate, which is issued by the DV_CERT_1_10. It has an expiration date BEFORE the effective date of T_CERT_1_1	
Version	1.0	
Referred by	ISO7816_J_11	
Content definition	See T_CERT_1	
Parameter	Certificate Authority Reference	See T_CERT_1
	Certificate Holder Reference	See T_CERT_1
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	Public Key reference	See T_CERT_1
	Signing Key reference	See T_CERT_1

2.5.2.39 T_CERT_1_3

Purpose	This certificate is an irregular Terminal certificate. This Terminal certificate is signed by the CVCA key.	
Version	1.0	
Referred by	ISO7816_J_38	
Content definition	See T_CERT_1	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	See T_CERT_1
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	See T_CERT_1
	Certificate expiration date	See T_CERT_1
	Public Key reference	See T_CERT_1
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.2.40 T_CERT_1_4

Purpose	This certificate is an irregular Terminal certificate. This signing key is an AT key.	
Version	1.0	
Referred by	ISO7816_J_47	
Content definition	See T_CERT_1	
Parameter	Certificate Authority Reference	DETESTTDE001
	Certificate Holder Reference	See T_CERT_1
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	See T_CERT_1
	Certificate expiration date	See T_CERT_1
	Public Key reference	See T_CERT_1
	Signing Key reference	Signed with the private key of key pair IS_KEY_01

2.5.2.41 T_CERT_1_5

Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_1 with shorter public Key (See DV_CERT_1_31)	
Version	1.0	
Referred by	ISO7816_J_49	
Content definition	See T_CERT_1	
Parameter	Certificate Authority Reference	See T_CERT_1
	Certificate Holder Reference	See T_CERT_1
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	See T_CERT_1
	Certificate expiration date	See T_CERT_1

	Public Key reference	See T_CERT_1
	Signing Key reference	Signed with the private key of key pair DV_KEY_01_2 (shorter than DV_KEY_01)

2.5.2.42 T_CERT_1_6

Purpose	This certificate is similar to T_CERT_1, but contains a Certificate Public Key with a short key. For RSA profile, same Algorithm Identifier but PK.DVCA's modulus length is shorter than the CVCA's key modulus length. For ECDSA profile, same Algorithm Identifier but DVCA's domain parameters are different and have a shorter prime length than the CVCA's key. The hash algorithm should be adapted if necessary.	
Version	1.0	
Referred by	ISO7816_J_50	
Content definition	See T_CERT_1	
Parameter	Certificate Authority Reference	See T_CERT_1
	Certificate Holder Reference	See T_CERT_1
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	See T_CERT_1
	Certificate expiration date	See T_CERT_1
	Public Key reference	Public key of key pair IS_KEY_01_1 (shorter than IS_KEY_01)
	Signing Key reference	Signed with the private key of key pair DV_KEY_01

2.5.3 Certificate Set 2

The certificate set follows a certification scheme for LDS1 application where the DV and IS permits access to data group 3 and/or 4.

2.5.3.1 DV_CERT_2_1

Purpose	This certificate is a regular DV certificate, with access rights for both data group 3 AND 4.	
Version	1.0	
Referred by	ISO7816_L_1, ISO7816_L_2, ISO7816_L_3, ISO7816_L_4	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE002
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_02
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.3.2 DV_CERT_2_2

Purpose	This certificate is a regular DV certificate, with access rights for data group 3 only.	
Version	1.0	
Referred by	ISO7816_L_5, ISO7816_L_6	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE002
	Certificate Holder Authorization	domestic DV, DG3
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_02
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.3.3 DV_CERT_2_3

Purpose	This certificate is a regular DV certificate, with access rights for data group 4 only.	
Version	1.0	
Referred by	ISO7816_L_7, ISO7816_L_8	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE002
	Certificate Holder Authorization	domestic DV, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_02

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.3.4 T_CERT_2_1

Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_2_1. It encodes access rights for data group 3 only.	
Version	1.0	
Referred by	ISO7816_L_1, ISO7816_L_2	
Parameter	Certificate Authority Reference	DETESTDVDE002
	Certificate Holder Reference	DETESTTDE002
	Certificate Holder Authorization	ePassport Application Inspection system, DG3
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair IS_KEY_02
	Signing Key reference	Signed with the private key of key pair DV_KEY_02

2.5.3.5 T_CERT_2_2

Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_2_1. It encodes access rights for data group 4 only.	
Version	1.0	
Referred by	ISO7816_L_3, ISO7816_L_4	
Parameter	Certificate Authority Reference	DETESTDVDE002
	Certificate Holder Reference	DETESTTDE002
	Certificate Holder Authorization	Inspection system, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair IS_KEY_02
	Signing Key reference	Signed with the private key of key pair DV_KEY_02

2.5.3.6 T_CERT_2_3

Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_2_2. It encodes access rights for data group 3 AND 4.	
Version	1.0	
Referred by	ISO7816_L_5, ISO7816_L_6 ISO7816_L_7, ISO7816_L_8	
Parameter	Certificate Authority Reference	DETESTDVDE002
	Certificate Holder Reference	DETESTTDE002
	Certificate Holder Authorization	Inspection system, DG3, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair IS_KEY_02
	Signing Key reference	Signed with the private key of key pair DV_KEY_02

2.5.4 Certificate Set 3

This certificate set contains certificate which have different effective and expiration dates to test the eMRTD behaviour for LDS1 application in respect to the update of the effective date and with expired certificates.

2.5.4.1 LINK_CERT_3_1

Note for ECDSA profile: Since the crypto mechanism is not changed by this link certificate it must be stated by the vendor of the test sample if the domain parameter should be included in this certificate (see ICS).

Purpose	This certificate is a link certificate which validity period starts one day before the original CVCA certificate expires.	
Version	1.0	
Referred by	ISO7816_M_3	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETEST_LINKDE3_1
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ - 1 day
	Certificate expiration date	$CVCA_{exp}$ + 2 month
	Public Key reference	Public key of key pair CVCA _KEY_03_1
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.4.2 LINK_CERT_3_2

Note for ECDSA profile: Since the crypto mechanism is not changed by this link certificate it must be stated by the vendor of the test sample if the domain parameter should be included in this certificate (see ICS).

Purpose	This certificate is a link certificate based on LINK_CERT_3_1	
Version	1.0	
Referred by	ISO7816_J_4	
Parameter	Certificate Authority Reference	DETEST_LINKDE3_1
	Certificate Holder Reference	DETEST_LINKDE3_2
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 1 month
	Certificate expiration date	$CVCA_{exp}$ + 4 month
	Public Key reference	Public key of key pair CVCA _KEY_03_2
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_03_1

2.5.4.3 DV_CERT_3_1

Purpose	This certificate is a domestic DV certificate, which validity period starts at the effective date of the CVCA and expires after one month	
Version	1.0	
Referred by	ISO7816_M_1, ISO7816_M_2	
Content definition	See DV_CERT_1 Tag 5F20 SHALL be consistent with Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE003
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_03
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.5.4.4 DV_CERT_3_2

Purpose	This certificate is similar to DV_CERT_3_1, but the certificate effective date is beyond the DV_CERT_3_1 expiration date.	
Version	1.0	
Referred by	ISO7816_M_2, ISO7816_M_3	
Content definition	See DV_CERT_3_1	
Parameter	Certificate Authority Reference	See DV_CERT_3_1
	Certificate Holder Reference	See DV_CERT_3_1
	Certificate Holder Authorization	See DV_CERT_3_1
	Certificate effective date	$CVCA_{eff}$ + 1 month + 1 day
	Certificate expiration date	$CVCA_{eff}$ + 2 month
	Public Key reference	See DV_CERT_3_1
	Signing Key reference	See DV_CERT_3_1

2.5.4.5 DV_CERT_3_3

Purpose	This certificate is a domestic DV certificate, which was issued by the update CVCA (LINK_CERT_3_1).	
Version	1.0	
Referred by	ISO7816_M_3	
Content definition	See DV_CERT_1 Tag 5F20 and 42 SHALL be consistent with Certificate Authority Reference and Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	DETEST_LINKDE3_1
	Certificate Holder Reference	DETESTDVDE003
	Certificate Holder Authorization	CVCA, All DGs in Read Access
	Certificate effective date	$CVCA_{exp}$ + 1 day
	Certificate expiration date	$CVCA_{exp}$ + 1 month

	Public Key reference	Public key of key pair DV_KEY_03
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_05_1

2.5.4.6 T_CERT_3_1

Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_3_1. This Terminal certificate has an advanced effective date. (Beyond the expiration date of T_CERT_3_2)	
Version	1.0	
Referred by	ISO7816_M_1	
Content definition	See T_CERT_1 Tag 5F20 and 42 SHALL be consistent with Certificate Authority Reference and Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	DETESTDVDE003
	Certificate Holder Reference	DETESTTDE003
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	$CVCA_{eff}$ + 14 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair IS_KEY_03
	Signing Key reference	Signed with the private key of key pair DV_KEY_03

2.5.4.7 T_CERT_3_2

Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_3_1. This Terminal certificate has an advanced expiration date Before the effective date of T_CERT_3_1	
Version	1.0	
Referred by	ISO7816_M_1	
Content definition	See T_CERT_3_1	
Parameter	Certificate Authority Reference	See T_CERT_3_1
	Certificate Holder Reference	See T_CERT_3_1
	Certificate Holder Authorization	See T_CERT_3_1
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	Public Key reference	See T_CERT_3_1
	Signing Key reference	See T_CERT_3_1

2.5.5 Certificate Set 4

This certificate set contains certificate which have different effective and expiration dates to test the eMRTD behaviour for LDS1 application in respect to the update of the effective date and with expired certificates. It completes Certificate Set 3

2.5.5.1 LINK_CERT_4_1

Note for ECDSA profile: Since the crypto mechanism is not changed by this link certificate it must be stated by the vendor of the test sample if the domain parameter should be included in this certificate (see ICS).

Purpose	This certificate is a link certificate based on LINK_CERT_3_2	
Version	1.0	
Referred by	ISO7816_M_4	
Parameter	Certificate Authority Reference	DETEST_LINKDE3_2
	Certificate Holder Reference	DETESTCADE004
	Certificate Holder Authorization	CVCA, DG3, DG4
	Certificate effective date	$CVCA_{exp}$ + 3 month
	Certificate expiration date	$CVCA_{exp}$ + 6 month
	Public Key reference	Public key of key pair CVCA _KEY_04_1
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_03_2

2.5.5.2 LINK_CERT_4_2

Note for ECDSA profile: Since the crypto mechanism is not changed by this link certificate it must be stated by the vendor of the test sample if the domain parameter should be included in this certificate (see ICS).

Purpose	This certificate is a link certificate based on LINK_CERT_4_1
Version	1.0

Referred by	ISO7816_M_5	
Parameter	Certificate Authority Reference	DETESTCADE004
	Certificate Holder Reference	DETEST_LINKDE4_2
	Certificate Holder Authorization	CVCA, DG3, DG4
	Certificate effective date	$CVCA_{exp}$ + 3 month + 10 days
	Certificate expiration date	$CVCA_{exp}$ + 8 month
	Public Key reference	Public key of key pair CVCA _KEY_04_2
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_04_1

2.5.5.3 DV_CERT_4

Purpose	This certificate is a domestic DV certificate, which was issued by the update CVCA (LINK_CERT_4_1).	
Version	1.0	
Referred by	ISO7816_M_4	
Content definition	See DV_CERT_1 Tag 5F20 and 42 SHALL be consistent with Certificate Authority Reference and Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	DETESTCADE004
	Certificate Holder Reference	DETESTDVDE004
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{exp}$ + 3 month
	Certificate expiration date	$CVCA_{exp}$ + 4 month
	Public Key reference	Public key of key pair DV_KEY_04
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_04_1

2.5.5.4 T_CERT_4

Purpose	This certificate is an irregular IS certificate. This signing key is a CVCA key.	
Version	1.0	
Referred by	ISO7816_M_5	
Content definition	See T_CERT_1 Tag 5F20 and 42 SHALL be consistent with Certificate Authority Reference and Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	DETEST_LINKDE4_2
	Certificate Holder Reference	DETESTTDE004
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	$CVCA_{exp}$ + 5 months
	Certificate expiration date	$CVCA_{exp}$ + 6 months
	Public Key reference	Public key of key pair IS_KEY_04
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_04_2

2.5.6 Certificate Set 5

This certificate set defines a link certificate used to update the chip signature mechanism according to the migration policy as defined by the manufacturer for LDS1 application. The cryptographic elements of these certificates SHALL use the new mechanisms besides the signature of the LINK_CERT_5 which is done with the original signature mechanism. This certificate set is only needed if the "Migration" profile is supported.

2.5.6.1 LINK_CERT_5

Note for ECDSA profile: Since the crypto mechanism is not changed by this link certificate it must be stated by the vendor of the test sample if the domain parameter should be included in this certificate (see ICS).

Purpose	**For MIG profile only:** This certificate is a link certificate, which defines a new crypto mechanism to be used by chip.	
Version	1.0	
Referred by	ISO7816_N_1	
Parameter	Certificate Authority Reference	DETESTCADE004
	Certificate Holder Reference	DETESTCADE005
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 7 months
	Certificate expiration date	$CVCA_{exp}$ + 10 months

	Public Key reference	Public key of key pair CVCA _KEY_05
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_04_1

2.5.6.2 DV_CERT_5

Purpose	**For MIG profile only:** This certificate is a domestic DV certificate, which was issued by the new CVCA (LINK_CERT_5).	
Version	1.0	
Referred by	ISO7816_N_1	
Content definition	See DV_CERT_1 Tag 5F20 and 42 SHALL be consistent with Certificate Authority Reference and Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	DETESTCADE005
	Certificate Holder Reference	DETESTDVDE005
	Certificate Holder Authorization	See DV_CERT_1
	Certificate effective date	$CVCA_{exp}$ + 7 months
	Certificate expiration date	$CVCA_{exp}$ + 8 months
	Public Key reference	Public key of key pair DV_KEY_05
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_05

2.5.6.3 T_CERT_5

Purpose	**For MIG profile only:** This certificate is a regular IS certificate, which was issued by DV_CERT_5.	
Version	1.0	
Referred by	ISO7816_N_1	
Content definition	See T_CERT_1 Tag 5F20 and 42 SHALL be consistent with Certificate Authority Reference and Certificate Holder reference as defined below	
Parameter	Certificate Authority Reference	DETESTDVDE005
	Certificate Holder Reference	DETESTTDE005
	Certificate Holder Authorization	See T_CERT_1
	Certificate effective date	$CVCA_{exp}$ + 7 months
	Certificate expiration date	$CVCA_{exp}$ + 8 months
	Public Key reference	Public key of key pair IS_KEY_05
	Signing Key reference	Signed with the private key of key pair DV_KEY_05

3 Security and Command Tests

3.1 Unit Test ISO7816_A – SelectApplication Command

3.1.1 Introduction

This test unit covers all tests about the SelectApplication command. The LDS specification requires the selection of the LDS application by its AID. Since the AID is unique, selecting the application SHOULD be possible regardless of the previously selected DF or EF. Selecting the LDS Application SHOULD also reset the cards security state but this scenario is tested in the access control unit test.

3.1.2 Test Case ISO7816_A_1

Purpose	Selecting the LDS Application using the AID (positive test)
Version	3.00
References	[R1] Part 11 4.3
Profile	BAC Only
Preconditions	LDS application SHALL NOT be selected.
Test scenario	1. Send the following SelectApplication APDU to the eMRTD. `'00 A4 04 0C 07 A0 00 00 02 47 10 01'`
Expected results	1. According to the ICAO recommendation, the P2 denotes "return no file information", and there is no Le byte present. Therefore, the response data field SHALL be empty. The eMRTD SHALL return status bytes '90 00'.
Postconditions	LDS application is selected.

3.1.3 Test Case ISO7816_A_2

Purpose	Selecting the LDS Application using the AID (positive test)
Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, PACE
Preconditions	LDS application SHALL NOT be selected.
Test scenario	1. Send the following SelectApplication APDU to the eMRTD. `'00 A4 04 0C 07 A0 00 00 02 47 10 01'`
Expected results	1. According to the ICAO recommendation, the P2 denotes "return no file information", and there is no Le byte present. Therefore, the response data field SHALL be empty. The eMRTD SHALL return status bytes '90 00'.
Postconditions	LDS application is selected.

3.1.4 Test Case ISO7816_A_3

Purpose	Selecting the LDS Application using the AID (positive test)
Version	3.00
References	[R1] Part 11 4.4.2
Profile	PACE Only
Preconditions	LDS application SHALL NOT be selected.
Test scenario	1. Send the following SelectApplication APDU to the eMRTD. `'00 A4 04 0C 07 A0 00 00 02 47 10 01'`
Expected results	1. According to the ICAO recommendation, the P2 denotes "return no file information", and there is no Le byte present. Therefore, the response data field SHALL be empty. The eMRTD SHALL return status bytes '90 00' or '6982'.
Postconditions	LDS application is selected if '90 00' is returned.

3.2 Unit Test ISO7816_B – Security conditions for BAC-protected eMRTDs

3.2.1 Introduction

This unit tests the security conditions of a BAC protected eMRTD. It SHALL NOT be possible read the content of any present file. The tests of this unit try to access the files with an explicit SelectFile command, a ReadBinary command with implicit file selection via the short file identifier (SFI), and unsecured ReadBinary while access is granted.
Note: Some eMRTDs allow selection of a protected file but no read access to this file.

The tests in this unit only apply to eMRTDs supporting BAC (profile BAC).

The tests in this unit do not test the secure messaging implementation including postconditions (e.g. SM termination); therefore, status bytes MAY be returned in secure messaging or without it. Unit ISO7816_C handles this. In the following test cases, "basic access is refused" means that protected data cannot be accessed. The term "basic access is granted" means that the inspection system has successfully authenticated to the eMRTD.

3.2.2 Test Case ISO7816_B_1

Purpose	Accessing the EF.COM file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. `'00 A4 02 0C 02 01 1E'`
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.3 Test Case ISO7816_B_2

Purpose	Accessing the EF.SOD file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. `'00 A4 02 0C 02 01 1D'`
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.4 Test Case ISO7816_B_3

Purpose	Accessing the EF.DG1 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. `'00 A4 02 0C 02 01 01'`
Expected results	1. The eMRTD SHALL return status bytes '69 82' or 90 00.
Postconditions	Preconditions remain unchanged.

3.2.5 Test Case ISO7816_B_4

Purpose	Accessing the EF.DG2 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.

Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 02'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.6 Test Case ISO7816_B_5

Purpose	Accessing the EF.DG3 file with explicit file selection
Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, DG3
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 03'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.7 Test Case ISO7816_B_6

Purpose	Accessing the EF.DG4 file with explicit file selection
Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, DG4
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 04'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.8 Test Case ISO7816_B_7

Purpose	Accessing the EF.DG5 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG5
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 05'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.9 Test Case ISO7816_B_8

Purpose	Accessing the EF.DG6 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG6
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 06'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.10 Test Case ISO7816_B_9

Purpose	Accessing the EF.DG7 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG7
Preconditions	The LDS application SHALL be selected.

Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 07'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.11 Test Case ISO7816_B_10

Purpose	Accessing the EF.DG8 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG8
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 08'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.12 Test Case ISO7816_B_11

Purpose	Accessing the EF.DG9 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG9
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 09'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.13 Test Case ISO7816_B_12

Purpose	Accessing the EF.DG10 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG10
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 0A'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.14 Test Case ISO7816_B_13

Purpose	Accessing the EF.DG11 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG11
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 0B'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.15 Test Case ISO7816_B_14

Purpose	Accessing the EF.DG12 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG12
Preconditions	The LDS application SHALL be selected.

Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 0C'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.16 Test Case ISO7816_B_15

Purpose	Accessing the EF.DG13 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG13
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 0D'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.17 Test Case ISO7816_B_16

Purpose	Accessing the EF.DG14 file with explicit file selection
Version	2.02
References	[R1] Part 11 4.3.2
Profile	BAC, (CA or PACE or AA-ECDSA)
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 0E'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.18 Test Case ISO7816_B_17

Purpose	Accessing the EF.DG15 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, AA
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 0F'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.19 Test Case ISO7816_B_18

Purpose	Accessing the EF.DG16 file with explicit file selection
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG16
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the following Select APDU to the eMRTD. '00 A4 02 0C 02 01 10'
Expected results	1. The eMRTD SHALL return status bytes '69 82' or '90 00'.
Postconditions	Preconditions remain unchanged.

3.2.20 Test Case ISO7816_B_19

Purpose	Accessing the EF.COM file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.

Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 9E 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.21 Test Case ISO7816_B_20

Purpose	Accessing the EF.SOD file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 9D 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.22 Test Case ISO7816_B_21

Purpose	Accessing the EF.DG1 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 81 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.23 Test Case ISO7816_B_22

Purpose	Accessing the EF.DG2 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 82 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.24 Test Case ISO7816_B_23

Purpose	Accessing the EF.DG3 file with implicit file selection (ReadBinary with SFI)
Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, DG3
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 83 00 00'
Expected results	1. Since read access is prohibited without BAC/EAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.25 Test Case ISO7816_B_24

Purpose	Accessing the EF.DG4 file with implicit file selection (ReadBinary with SFI)

Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, DG4
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 84 00 00'
Expected results	1. Since read access is prohibited without BAC/EAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.26 Test Case ISO7816_B_25

Purpose	Accessing the EF.DG5 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG5
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 85 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.27 Test Case ISO7816_B_26

Purpose	Accessing the EF.DG6 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG6
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 86 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.28 Test Case ISO7816_B_27

Purpose	Accessing the EF.DG7 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG7
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 87 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.29 Test Case ISO7816_B_28

Purpose	Accessing the EF.DG8 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG8
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 88 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.

Postconditions	Preconditions remain unchanged.

3.2.30 Test Case ISO7816_B_29

Purpose	Accessing the EF.DG9 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG9
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. `'00 B0 89 00 00'`
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.31 Test Case ISO7816_B_30

Purpose	Accessing the EF.DG10 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG10
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. `'00 B0 8A 00 00'`
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.32 Test Case ISO7816_B_31

Purpose	Accessing the EF.DG11 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG11
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. `'00 B0 8B 00 00'`
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.33 Test Case ISO7816_B_32

Purpose	Accessing the EF.DG12 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG12
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. `'00 B0 8C 00 00'`
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.34 Test Case ISO7816_B_33

Purpose	Accessing the EF.DG13 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG13
Preconditions	The LDS application SHALL be selected.

Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 8D 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.35 Test Case ISO7816_B_34

Purpose	Accessing the EF.DG14 file with implicit file selection (ReadBinary with SFI)
Version	2.02
References	[R1] Part 11 4.3.2
Profile	BAC, (CA or PACE or AA-ECDSA)
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 8E 00 00'
Expected results	1. Since read access is prohibited without BAC or PACE, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.36 Test Case ISO7816_B_35

Purpose	Accessing the EF.DG15 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, AA
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 8F 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.37 Test Case ISO7816_B_36

Purpose	Accessing the EF.DG16 file with implicit file selection (ReadBinary with SFI)
Version	1.1
References	[R1] Part 11 4.3.2
Profile	BAC, DG16
Preconditions	The LDS application SHALL be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD. '00 B0 90 00 00'
Expected results	1. Since read access is prohibited without BAC, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82'.
Postconditions	Preconditions remain unchanged.

3.2.38 Test Case ISO7816_B_37

Purpose	Accessing the EF.COM file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.COM encoded as a valid SM APDU to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.

	2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.39 Test Case ISO7816_B_38

Purpose	Accessing the EF. SOD file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.SOD encoded as a valid SM APDU to the eMRTD. '0C B0 9D 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.40 Test Case ISO7816_B_39

Purpose	Accessing the EF. DG1 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG1 encoded as a valid SM APDU to the eMRTD. '0C B0 81 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.41 Test Case ISO7816_B_40

Purpose	Accessing the EF. DG2 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG2 encoded as a valid SM APDU to the eMRTD. '0C B0 82 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.42 Test Case ISO7816_B_41

Purpose	Accessing the EF. DG3 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.

	This test is not applicable if an extended access control or a proprietary protocol is supported to protect sensitive data.
Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, DG3
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG3 encoded as a valid SM APDU to the eMRTD. '0C B0 83 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.43 Test Case ISO7816_B_42

Purpose	Accessing the EF. DG4 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted. This test is not applicable if an extended access control or a proprietary protocol is supported to protect sensitive data.
Version	3.00
References	[R1] Part 11 4.3.2
Profile	BAC, DG4
Preconditions	The LDS application SHALL be selected and basic (extended) access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG4 encoded as a valid SM APDU to the eMRTD. '0C B0 84 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.44 Test Case ISO7816_B_43

Purpose	Accessing the EF. DG5 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG5
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG5 encoded as a valid SM APDU to the eMRTD. '0C B0 85 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.45 Test Case ISO7816_B_44

Purpose	Accessing the EF. DG6 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04

References	[R1] Part 11 4.3.2
Profile	BAC, DG6
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG6 encoded as a valid SM APDU to the eMRTD. '0C B0 86 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.46 Test Case ISO7816_B_45

Purpose	Accessing the EF. DG7 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG7
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG7 encoded as a valid SM APDU to the eMRTD. '0C B0 87 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.47 Test Case ISO7816_B_46

Purpose	Accessing the EF. DG8 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG8
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG8 encoded as a valid SM APDU to the eMRTD. '0C B0 88 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.48 Test Case ISO7816_B_47

Purpose	Accessing the EF. DG9 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG9
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG9 encoded as a valid SM APDU to the eMRTD. '0C B0 89 00 0D 97 01 06 8E 08 <checksum> 00'

	2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.49 Test Case ISO7816_B_48

Purpose	Accessing the EF. DG10 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG10
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG10 encoded as a valid SM APDU to the eMRTD. '0C B0 8A 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.50 Test Case ISO7816_B_49

Purpose	Accessing the EF. DG11 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG11
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG11 encoded as a valid SM APDU to the eMRTD. '0C B0 8B 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.51 Test Case ISO7816_B_50

Purpose	Accessing the EF. DG12 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG12
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG12 encoded as a valid SM APDU to the eMRTD. '0C B0 8C 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.52 Test Case ISO7816_B_51

Purpose	Accessing the EF. DG13 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, DG13
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG13 encoded as a valid SM APDU to the eMRTD. '0C B0 8D 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.53 Test Case ISO7816_B_52

Purpose	Accessing the EF. DG14 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, (CA or PACE or AA-ECDSA)
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG14 encoded as a valid SM APDU to the eMRTD. '0C B0 8E 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.54 Test Case ISO7816_B_53

Purpose	Accessing the EF. DG15 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2
Profile	BAC, AA
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG15 encoded as a valid SM APDU to the eMRTD. '0C B0 8F 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.2.55 Test Case ISO7816_B_54

Purpose	Accessing the EF. DG16 file with ReadBinary. The test verifies the enforcement of Secure Messaging while basic access is granted.
Version	2.04
References	[R1] Part 11 4.3.2

Profile	BAC, DG16
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG16 encoded as a valid SM APDU to the eMRTD. '0C B0 '90 00' 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following ReadBinary APDU as a plain unprotected APDU to the eMRTD. '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.3 Unit Test ISO7816_C – Basic Access Control

3.3.1 Introduction

This unit checks the BAC implementation of the eMRTD. The complete BAC access mechanism is tested, including robustness tests with invalid input data.

In the following test cases, “basic access is refused” means that there are no valid session keys for secure messaging available and that access to any BAC protected file is refused. The term “basic access is granted” means that the inspection system has successfully authenticated to the eMRTD and that valid session keys are available for secure messaging.

The READ BINARY command in SM mode is used in the following test cases to verify that the session keys are no longer valid. Alternatively, the command SELECT FILE in SM mode MAY be used.

3.3.2 Test Case ISO7816_C_1

Purpose	This function verifies the GetChallenge command (positive test).
Version	1.1
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be refused.
Test scenario	1. Send the following GetChallenge APDU to the eMRTD. `'00 84 00 00 08'` 2. Send the same GetChallenge APDU to the eMRTD. `'00 84 00 00 08'`
Expected results	1. The eMRTD SHALL return 8 random bytes and the status bytes '90 00'. 2. The eMRTD SHALL return 8 different random bytes and the status bytes '90 00'.
Postconditions	Preconditions remain unchanged.

3.3.3 Test Case ISO7816_C_2

Purpose	This test checks the response to the MutualAuthenticate command (positive test).
Version	1.1
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be refused.

Test scenario	1. Send the following GetChallenge APDU to the eMRTD.
	`'00 84 00 00 08'`
	2. Send the MutualAuthenticate APDU to the eMRTD. The <data> SHALL be calculated from the given MRZ data and the challenge returned in step 1.
	`'00 82 00 00 28 <data> 28'`

Expected results	1. The eMRTD SHALL return 8 random bytes and the status bytes '90 00'. 2. The response from the eMRTD SHALL be verified as specified in [R1]. The returned status bytes SHALL be '90 00'.
Postconditions	Basic access is granted.

3.3.4 Test Case ISO7816_C_3

Purpose	This test checks the authentication failure response to the MutualAuthenticate command
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be refused.
Test scenario	1. Send the following GetChallenge APDU to the eMRTD. `'00 84 00 00 08'` 2. Send the MutualAuthenticate APDU to the eMRTD. Same as ISO7816_C_2, but for the <data> calculation data from a different MRZ

	SHALL be used. To achieve this, the document number SHALL be increment by 1 before the <data> is calculated. '00 82 00 00 28 <data> 28'
Expected results	1. The eMRTD SHALL return 8 random bytes and the status bytes '90 00'. 2. The eMRTD SHALL respond with an ISO warning or ISO checking error or ISO execution error.
Postconditions	Preconditions remain unchanged.

3.3.5 Test Case ISO7816_C_4

Purpose	This test checks the authentication failure response to the MutualAuthenticate command
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be refused. The GetChallenge command SHALL NOT have been executed.
Test scenario	1. Send the MutualAuthenticate APDU to the eMRTD. Same as ISO7816_C_2, but for the <data> calculation the challenge '00 00 00 00 00 00 00 00' SHALL be used. '00 82 00 00 28 <data> 28' 2. Send the following GetChallenge APDU to the eMRTD. '00 84 00 00 08' 3. Send the following GetChallenge APDU to the eMRTD. '00 84 00 00 08' 4. Send the MutualAuthenticate APDU to the eMRTD. Same as ISO7816_C_2, but for the <data> calculation the challenge of step 2 SHALL be used. '00 82 00 00 28 <data> 28'
Expected results	1. The eMRTD SHALL respond with an ISO warning or ISO checking error or ISO execution error. 2. The eMRTD SHALL return 8 random bytes and the status bytes '90 00'. 3. The eMRTD SHALL return 8 random bytes and the status bytes '90 00'. 4. The eMRTD SHALL respond with an ISO warning or ISO checking error or ISO execution error.
Postconditions	Preconditions remain unchanged.

3.3.6 Void

Removed in version 3.00.

3.3.7 Test Case ISO7816_C_6

Purpose	This test checks the response for the MutualAuthenticate command with a corrupted MAC.
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be refused.
Test scenario	1. Send the following GetChallenge APDU to the eMRTD. '00 84 00 00 08' 2. Send the MutualAuthenticate APDU to the eMRTD. The <data> SHALL be calculated from the given MRZ data and the challenge returned in step 1. In the calculated MAC the very last byte is incremented by one. '00 82 00 00 28 <data> 28'
Expected results	1. The eMRTD SHALL return 8 random bytes and the status bytes '90 00'. 2. The eMRTD SHALL respond with an ISO warning or ISO checking error or ISO execution error.
Postconditions	Preconditions remain unchanged.

Note: this test case differs from test case ISO7816_C_3. In this test case, only the MAC is manipulated but the cryptogram is valid.

3.3.8 Void

3.3.9 Test Case ISO7816_C_8

Purpose	This test checks the Secure Messaging coding of a ReadBinary (B0) with SFI (positive tests)
Version	1.1
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following ReadBinary (SFI) APDU encoded as a valid SM APDU to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00' 2. Search for the cryptogram DO encoded in tag '87' and decrypt it with current session key. 3. Search for the processing status DO encoded in tag '99' and verify status bytes received. 4. Search for the cryptographic checksum DO encoded in tag '8E' and verify it with current session key.
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The response of step 1 SHALL contain the read data in a valid cryptogram encoded in tag '87'. 3. The response of step 1 SHOULD contain SW encoded in tag '99' that equals the status bytes of the secured response. 4. The response of step 1 SHALL contain a valid cryptographic checksum encoded in tag '8E'.
Postconditions	Preconditions remain unchanged.

3.3.10 Test Case ISO7816_C_9

Purpose	This test checks the Secure Messaging coding of a ReadBinary ('B1') with SFI (positive tests)
Version	1.1
References	[R1] Part 11 4.3 [R3] for TLV encoded data objects
Profile	BAC, OddIns
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following ReadBinary (SFI) APDU encoded as a valid SM APDU to the eMRTD. The offset (0) SHALL be encoded in a DO '54', which is then encrypted in a SM '85' object. '0C B1 00 1E 17 85 08 <cryptogram> 97 01 06 8E 08 <checksum> 00' 2. Search for the cryptogram DO encoded in tag '85' and decrypt it with current session key. 3. Search for the processing status DO encoded in tag '99' and verify status bytes received. 4. Search for the cryptographic checksum DO encoded in tag '8E' and verify it with current session key.
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The response of step 1 SHALL contain the read data in a valid cryptogram encoded in tag '85'. The data SHALL be encapsulated in a tag '53' object. 3. The response of step 1 SHOULD contain SW encoded in tag '99' that equals the status bytes of the secured response. 4. The response of step 1 SHALL contain a valid cryptographic checksum encoded in tag '8E'.
Postconditions	Preconditions remain unchanged.

3.3.11 Test Case ISO7816_C_10

Purpose	This test checks the Secure Messaging coding of a SelectFile and ReadBinary (B0) w/o SFI (positive tests)
Version	1.1
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU encoded as a valid SM APDU to the eMRTD. '0C A4 02 0C 15 87 09 01 <cryptogram> 8E 08 <checksum> 00' 2. Search for the processing status DO encoded in tag '99' and verify status bytes received. 3. Search for the cryptographic checksum DO encoded in tag '8E' and verify it with current session key. 4. Send the following ReadBinary APDU encoded as a valid SM APDU to the eMRTD. '0C B0 00 00 0D 97 01 06 8E 08 <checksum> 00' 5. Search for the cryptogram DO encoded in tag '87' and decrypt it with current session key. 6. Search for the processing status DO encoded in tag '99' and verify status bytes received. 7. Search for the cryptographic checksum DO encoded in tag '8E' and verify it with current session key. 8. Search for further DO.
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The response of step 1 SHALL contain SW encoded in tag '99' that SHALL equal the received status bytes of the secured response. 3. The response of step 1 SHALL contain a valid cryptographic checksum encoded in tag '8E'. 4. The eMRTD SHALL return the status bytes '90 00'. 5. The response of step 4 SHALL contain the read data in a valid cryptogram encoded in tag '87'. 6. The response of step 4 SHOULD contain SW encoded in tag '99' that equals the received status bytes of the secured response. 7. The response of step 4 SHALL contain a valid cryptographic checksum encoded in tag '8E'. 8. The response SHALL NOT contain any further data but the response trailer.
Postconditions	Preconditions remain unchanged.

3.3.12 Test Case ISO7816_C_11

Purpose	This test checks the Secure Messaging coding of a SelectFile and ReadBinary (B1) w/o SFI (positive tests)
Version	1.1
References	[R1] Part 11 4.3 [R3] for TLV encoded data objects
Profile	BAC, OddIns
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU encoded as a valid SM APDU to the eMRTD. '0C A4 02 0C 15 87 09 01 <cryptogram> 8E 08 <checksum> 00' 2. Search for the processing status DO encoded in tag '99' and verify status bytes received.

	3. Search for the cryptographic checksum DO encoded in tag 8E and verify it with current session key. 4. Send the following ReadBinary APDU encoded as a valid SM APDU to the eMRTD. The offset (0) SHALL be encoded in a DO '54', which is then encrypted in a SM '85' object. '0C B1 00 00 17 85 08 <cryptogram> 97 01 06 8E 08 <checksum> 00' 5. Search for the cryptogram DO encoded in tag '85' and decrypt it with current session key. 6. Search for the processing status DO encoded in tag '99' and verify status bytes received. 7. Search for the cryptographic checksum DO encoded in tag '8E' and verify it with current session key. 8. Search for further DO.
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The response of step 1 SHALL contain SW encoded in tag '99' that SHALL equal the received status bytes of the secured response. 3. The response of step 1 SHALL contain a valid cryptographic checksum encoded in tag '8E'. 4. The eMRTD SHALL return the status bytes '90 00'. 5. The response of step 4 SHALL contain the read data in a valid cryptogram encoded in tag '85'. The data SHALL be encapsulated in a tag '53' object. 6. The response of step 4 SHOULD contain SW encoded in tag '99' that equals the received status bytes of the secured response. 7. The response of step 4 SHALL contain a valid cryptographic checksum encoded in tag '8E'. 8. The response SHALL NOT contain any further data but the response trailer.
Postconditions	Preconditions remain unchanged.

3.3.13 Test Case ISO7816_C_12

Purpose	The test verifies the Secure Messaging handling while basic access is granted for the SelectFile Command (checksum missing)
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU encoded as a SM APDU but without the checksum SM object to the eMRTD. '0C A4 02 0C 0B 87 09 01 <cryptogram> 00' 2. To verify that the error in step 1 has terminated the SM session, send a valid SM APDU (ReadBinary) to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error. 2. Since the session keys are no longer valid, the eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.3.14 Test Case ISO7816_C_13

Purpose	The test verifies the Secure Messaging handling while basic access is granted for the SelectFile Command (checksum corrupted)
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU encoded

	as a valid SM APDU to the eMRTD. The last byte of the checksum is incremented by one. '0C A4 02 0C 15 87 09 01 <cryptogram> 8E 08 <corrupted checksum> 00' 2. To verify that the error in step 1 has terminated the SM session, send a valid SM APDU (ReadBinary) to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return status bytes '69 88' or '69 82'. 2. Since the session keys are no longer valid, the eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.3.15 Test Case ISO7816_C_14

Purpose	The test verifies the Secure Messaging handling while basic access is granted for the SelectFile Command (bad send sequence counter)
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU encoded as a valid SM APDU to the eMRTD. During the coding of the SM APDU the SendSequenceCounter is not incremented. '0C A4 02 0C 15 87 09 01 <cryptogram> 8E 08 <corrupted checksum> 00' 2. To verify that the error in step 1 has terminated the SM session, send a valid SM APDU (ReadBinary) to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return status bytes '69 88' or '69 82'. 2. Since the session keys are no longer valid, the eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.3.16 Void

Removed in version 2.11.

3.3.17 Test Case ISO7816_C_16

Purpose	The test verifies the enforcement of Secure Messaging while basic access is granted for the SelectFile Command.
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. EF.COM SHALL be selected. Send the following SelectFile APDU as a plain unprotected APDU to the eMRTD. '00 A4 02 0C 02 01 1E'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or status bytes '90 00'.
Postconditions	Postcondition depends on the status bytes returned by the eMRTD.

3.3.18 Test Case ISO7816_C_17

Purpose	The test verifies the Secure Messaging handling while basic access is granted for the ReadBinary Command (checksum missing).
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.

Test scenario	1. Send the following ReadBinary APDU encoded as a SM APDU but without the checksum SM object to the eMRTD. '0C B0 9E 00 03 97 01 06 00' 2. To verify that the error in step 1 has terminated the SM session, send a valid SM APDU (ReadBinary) to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error. 2. Since the session keys are no longer valid, the eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.3.19 Test Case ISO7816_C_18

Purpose	The test verifies the Secure Messaging handling while basic access is granted for the ReadBinary Command (checksum corrupted).
Version	2.04
References	[R1] Part 11 4.3
Profile	BAC
Preconditions	The LDS application SHALL be selected and basic access SHALL be granted.
Test scenario	1. Send the following ReadBinary APDU encoded as a valid SM APDU to the eMRTD. The last byte of the checksum is incremented by one. '0C B0 00 00 0D 97 01 06 8E 08 <checksum> 00' 2. To verify that the error in step 1 has terminated the SM session, send a valid SM APDU (ReadBinary) to the eMRTD. '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return status bytes '69 88' or '69 82'. 2. Since the session keys are no longer valid, the eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Basic access is refused.

3.3.20 Void

Removed in version 2.11

3.4 Unit Test ISO7816_D – Protected SelectFile Command

3.4.1 Introduction

This unit verifies the implementation of the protected SelectFile command.

The data groups are accessible after the “Open LDS Application" (see subclause 2.1) procedure has been performed.
All test cases of this test unit which require the “Open LDS Application” procedure SHALL be performed twice (one test run with BAC and one with PACE) if the chip supports both protocols. If the chip only supports one of these protocols (BAC or PACE), only one test run has to be performed with the supported protocol used in the “Open LDS Application” procedure.

The tests in this unit do not test the secure messaging implementation including postconditions (e.g. SM termination); therefore, status bytes MAY be returned in secure messaging or without it. Unit ISO7816_C handles this for BAC and Unit ISO7816_P handles this for PACE.

Note: when accessing to sensitive data protected by Extended Access control, EAC SHALL be granted.

3.4.2 Test Case ISO7816_D_1

Purpose	This test case verifies the SelectFile (EF.COM) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.COM is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte 60 and the status bytes '90 00'.
Postconditions	EF.COM SHALL be selected.

3.4.3 Void

Removed in version 2.11.

3.4.4 Void

Removed in version 3.00.

3.4.5 Void

Removed in version 3.00.

3.4.6 Void

Removed in version 3.00.

3.4.7 Test Case ISO7816_D_6

Purpose	This test case verifies the SelectFile (EF.SOD) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.SOD SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier 01 1D. Send the following SelectFile APDU to the eMRTD.

	'0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.SOD is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '77' and the status bytes '90 00'.
Postconditions	EF.SOD is selected.

3.4.8 Test Case ISO7816_D_7

Purpose	This test case verifies the SelectFile (EF.DG1) command (positive test).
Version	or PACE
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG1 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 01'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG1is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '61' and the status bytes '90 00'.
Postconditions	EF.DG1 is selected.

3.4.9 Test Case ISO7816_D_8

Purpose	This test case verifies the SelectFile (EF.DG2) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG2 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 02'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG2 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '75' and the status bytes '90 00'.
Postconditions	EF.DG2 is selected.

3.4.10 Test Case ISO7816_D_9

Purpose	This test case verifies the SelectFile (EF.DG3) command (positive test).
Version	2.07
References	[R1] Part 10 3.5
Profile	((BAC or PACE), DG3)
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG3 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 03'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00'

	2. To verify that EF.DG3 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '63' and the status bytes '90 00'.
Postconditions	EF.DG3 is selected.

3.4.11 Test Case ISO7816_D_10

Purpose	This test case verifies the SelectFile (EF.DG4) command (positive test).
Version	2.07
References	[R1] Part 10 3.5
Profile	((BAC or PACE), DG4)
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG4 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 04'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG4 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '76' and the status bytes '90 00'.
Postconditions	EF.DG4 is selected.

3.4.12 Test Case ISO7816_D_11

Purpose	This test case verifies the SelectFile (EF.DG5) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG5
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG5 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 05'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG5 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '65' and the status bytes '90 00'.
Postconditions	EF.DG5 is selected.

3.4.13 Test Case ISO7816_D_12

Purpose	This test case verifies the SelectFile (EF.DG6) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG6
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG6 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 06'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG6 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'

Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '66' and the status bytes '90 00'.
Postconditions	EF.DG6 is selected.

3.4.14 Test Case ISO7816_D_13

Purpose	This test case verifies the SelectFile (EF.DG7) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG7
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG7 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 07'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG7 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '67' and the status bytes '90 00'.
Postconditions	EF.DG7 is selected.

3.4.15 Test Case ISO7816_D_14

Purpose	This test case verifies the SelectFile (EF.DG8) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG8
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG8 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 08'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG8 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '68' and the status bytes '90 00'.
Postconditions	EF.DG8 is selected.

3.4.16 Test Case ISO7816_D_15

Purpose	This test case verifies the SelectFile (EF.DG9) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG9
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG9 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 09'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG9 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '69' and the status bytes '90 00'.
Postconditions	EF.DG9 is selected.

3.4.17 Test Case ISO7816_D_16

Purpose	This test case verifies the SelectFile (EF.DG10) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG10
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG10 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 0A'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG10 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '6A' and the status bytes '90 00'.
Postconditions	EF.DG10 is selected.

3.4.18 Test Case ISO7816_D_17

Purpose	This test case verifies the SelectFile (EF.DG11) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG11
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG11 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 0B'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG11 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '6B' and the status bytes '90 00'.
Postconditions	EF.DG11 is selected.

3.4.19 Test Case ISO7816_D_18

Purpose	This test case verifies the SelectFile (EF.DG12) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG12
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG12 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 0C'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG12 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '6C' and the status bytes '90 00'.
Postconditions	EF.DG12 is selected.

3.4.20 Test Case ISO7816_D_19

Purpose	This test case verifies the SelectFile (EF.DG13) command (positive test).

Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG13
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG13 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 0D'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG13 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '6D' and the status bytes '90 00'.
Postconditions	EF.DG13 is selected.

3.4.21 Test Case ISO7816_D_20

Purpose	This test case verifies the SelectFile (EF.DG14) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC , (EAC or PACE or AA-ECDSA)
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG14 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 0E'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG14 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '6E' and the status bytes '90 00'.
Postconditions	EF.DG14 is selected.

3.4.22 Test Case ISO7816_D_21

Purpose	This test case verifies the SelectFile (EF.DG15) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), AA
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG15 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 0F'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG15 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '6F' and the status bytes '90 00'.
Postconditions	File EF.DG15 is selected.

3.4.23 Test Case ISO7816_D_22

Purpose	This test case verifies the SelectFile (EF.DG16) command (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG16

Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. EF.DG16 SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 10'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. To verify that EF.DG16 is selected, send a valid ReadBinary APDU to the eMRTD. '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return byte '70' and the status bytes '90 00'.
Postconditions	EF.DG16 is selected.

3.4.24 Test Case ISO7816_D_23

Purpose	This test case verifies the SelectFile command when the file to be selected does not exist.
Version	2.04
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. A not existing file SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '02 02'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Preconditions remain unchanged.

3.5 Unit Test ISO7816_E – Protected ReadBinary Command

3.5.1 Introduction

This unit verifies the implementation of the protected ReadBinary command.

The data groups are accessible after the “Open LDS Application" (see subclause 2.1) procedure has been performed.
All test cases of this test unit which require the “Open LDS Application” procedure SHALL be performed twice (one test run with BAC and one with PACE) if the chip supports both protocols. If the chip only supports one of these protocols (BAC or PACE), only one test run has to be performed with the supported protocol used in the “Open LDS Application” procedure.

The tests in this unit do not test the secure messaging implementation including postconditions (e.g. SM termination); therefore, status bytes MAY be returned in secure messaging or without it. Unit ISO7816_C handles this for BAC and Unit ISO7816_P handles this for PACE.

Note: For the ReadBinary command in Secure Messaging mode, there is no clear definition in the ISO specification if the Le byte in DO '97' = '00'. Test cases E_1 to E_3 and E_5 to E_22 use Le = '01' in order to avoid unspecified EOF situations

Note: when accessing to sensitive data protected by Extended Access Control, EAC SHALL be granted.

3.5.2 Test Case ISO7816_E_1

Purpose	This test case verifies the ReadBinary command (w/o SFI) (positive test).
Version	2.04
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. Send the ReadBinary APDU to the eMRTD and read the first bytes of EF.COM '0C B0 00 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return response data '60' and the status bytes '90 00'.
Postconditions	Preconditions remain unchanged.

3.5.3 Void

Removed in version 2.11.

3.5.4 Test Case ISO7816_E_3

Purpose	This test case verifies the ReadBinary command (w/o SFI) (offset beyond EOF).
Version	2.04
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. This test case implicitly tests the SelectFile command; so it is required that the eMRTD has previously passed the SelectFile Test ISO7816_D_1, otherwise this test will fail.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L_{87}> 01 <cryptogram> 8E 08 <checksum> 00'

	2. Send the ReadBinary APDU to the eMRTD. The offset is beyond the end of the EF.COM file. Note: Since the actual file on the eMRTD could be larger than necessary, the eMRTD may return valid data in this case. If this happens, the test may have to be repeated with an appropriated offset. '0C B0 7F FF 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return an ISO checking error or ISO execution error.
Postconditions	Preconditions remain unchanged.

3.5.5 Test Case ISO7816_E_4

Purpose	This test case verifies the ReadBinary command (w/o SFI) (Le beyond EOF).
Version	2.04
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. This test case implicitly tests the SelectFile command; so it is required that the eMRTD has previously passed the SelectFile Test ISO7816_D_1, otherwise this test will fail.
Test scenario	1. EF.COM SHALL be selected. Therefore, the cryptogram SHALL contain the file identifier '01 1E'. Send the following SelectFile APDU to the eMRTD. '0C A4 02 0C <Lc> 87 <L87> 01 <cryptogram> 8E 08 <checksum> 00' 2. Send the ReadBinary APDU to the eMRTD. The Le Byte requests more data than available in the EF.COM file Note: Since the actual file on the eMRTD could be larger than necessary, the eMRTD may return valid data in this case. If this happens, the test may have to be repeated with an appropriated offset. '0C B0 00 00 0D 97 01 E0 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'. 2. The eMRTD SHALL return status bytes '90 00' or an ISO warning or an ISO checking error or ISO execution error.
Postconditions	Preconditions remain unchanged.

3.5.6 Test Case ISO7816_E_5

Purpose	This test case verifies the ReadBinary command (EF.COM SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.COM. '0C B0 9E 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.COM is selected.

3.5.7 Test Case ISO7816_E_6

Purpose	This test case verifies the ReadBinary command (EF.SOD SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.SOD. '0C B0 9D 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.SOD is selected.

3.5.8 Test Case ISO7816_E_7

Purpose	This test case verifies the ReadBinary command (EF.DG1 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. EF.DG1 SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG1. '0C B0 81 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG1 is selected.

3.5.9 Test Case ISO7816_E_8

Purpose	This test case verifies the ReadBinary command (EF.DG2 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC or PACE
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG2. '0C B0 82 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG2 is selected.

3.5.10 Test Case ISO7816_E_9

Purpose	This test case verifies the ReadBinary command (EF.DG3 SFI) (positive test).
Version	2.07
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG3
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG3. '0C B0 83 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG3 is selected.

3.5.11 Test Case ISO7816_E_10

Purpose	This test case verifies the ReadBinary command (EF.DG4 SFI) (positive test).
Version	2.07
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG4
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG3. '0C B0 84 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG4 is selected.

3.5.12 Test Case ISO7816_E_11

Purpose	This test case verifies the ReadBinary command (EF.DG5 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG5
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.

Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG5. '0C B0 85 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG5 is selected.

3.5.13 Test Case ISO7816_E_12

Purpose	This test case verifies the ReadBinary command (EF.DG6 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG6
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG6. '0C B0 86 00 0D 97 01 E0 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG6 is selected.

3.5.14 Test Case ISO7816_E_13

Purpose	This test case verifies the ReadBinary command (EF.DG7 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG7
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG7. '0C B0 87 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG7 is selected.

3.5.15 Test Case ISO7816_E_14

Purpose	This test case verifies the ReadBinary command (EF.DG8 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG8
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG8. '0C B0 88 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG8 is selected.

3.5.16 Test Case ISO7816_E_15

Purpose	This test case verifies the ReadBinary command (EF.DG9 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG9
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG9. '0C B0 89 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG9 is selected.

3.5.17 Test Case ISO7816_E_16

Purpose	This test case verifies the ReadBinary command (EF.DG10 SFI) (positive test).

Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG10
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG10. '0C B0 8A 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG10 is selected.

3.5.18 Test Case ISO7816_E_17

Purpose	This test case verifies the ReadBinary command (EF.DG11 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG11
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG11. '0C B0 8B 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG11 is selected.

3.5.19 Test Case ISO7816_E_18

Purpose	This test case verifies the ReadBinary command (EF.DG12 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG12
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG12. '0C B0 8C 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG12 is selected.

3.5.20 Test Case ISO7816_E_19

Purpose	This test case verifies the ReadBinary command (EF.DG13 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG13
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG13. '0C B0 8D 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG13 is selected.

3.5.21 Test Case ISO7816_E_20

Purpose	This test case verifies the ReadBinary command (EF.DG14 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	BAC, (EAC or PACE or AA-ECDSA)
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG14. '0C B0 8E 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.

Postconditions	EF.DG14 is selected.

3.5.22 Test Case ISO7816_E_21

Purpose	This test case verifies the ReadBinary command (EF.DG15 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), AA
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG15. '0C B0 8F 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG15 is selected.

3.5.23 Test Case ISO7816_E_22

Purpose	This test case verifies the ReadBinary command (EF.DG16 SFI) (positive test).
Version	2.02
References	[R1] Part 10 3.5
Profile	(BAC or PACE), DG16
Preconditions	The LDS application SHALL be selected. An EF SHALL NOT be selected.
Test scenario	1. Send the ReadBinary APDU to the eMRTD, this will read the first byte of the EF.DG16. '0C B0 '90 00' 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return the status bytes '90 00'.
Postconditions	EF.DG16 is selected.

3.6 Unit Test ISO7816_H – Security conditions for EAC-protected eMRTDs

3.6.1 Introduction

The eMRTDs containing sensitive biometric data SHALL be protected by the terminal authentication mechanism. While all other data groups are accessible after the "Open LDS Application" (see subclause 2.1) procedure has been performed, the data group 3 and/or 4 SHALL only be accessible after a successful terminal authentication process.
All test cases of this test unit which require the "Open LDS Application" procedure SHALL be performed twice (one test run with BAC and one with PACE) if the chip supports both protocols. If the chip only supports one of these protocols (BAC or PACE), only one test run has to be performed with the supported protocol used in the "Open LDS Application" procedure.

Note: Chip Authentication must be performed as described in [R1] Part 11 §6.2 either with command MSE:Set KAT or commands MSE:Set AT and General Authenticate.

3.6.2 Test case ISO7816_H_1

Purpose	SELECT command for EF.DG3 within an established PACE or BAC session, but before the terminal authentication mechanism has been performed.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG3
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given SELECT APDU for EF.DG3 (File Id '01 03') to the eMRTD. Though PACE or BAC and the CA mechanisms have been performed, the access to the data group 3 SHALL be denied. '0C A4 02 0C 15 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the encrypted file ID ('01 03'). 2. Send the following READ BINARY command to the eMRTD: '0C B0 00 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. ISO checking or ISO execution error '90 00' within a valid SM response. If this step returns an ISO checking or ISO execution error, the next step SHALL be skipped. 2. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.3 Test case ISO7816_H_2

Purpose	SELECT command for EF.DG4 within an established PACE or BAC session, but before the terminal authentication mechanism has been performed.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG4
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given SELECT APDU for EF.DG4 (File Id '01 04') to the eMRTD. Though PACE or BAC and the CA mechanisms have been performed, the access to the data group 4 SHALL be denied. '0C A4 02 0C 15 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the encrypted file ID ('01 04'). 2. Send the following READ BINARY command to the eMRTD: '0C B0 00 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. ISO checking or ISO execution error '90 00' within a valid SM response. If this step returns an ISO checking or ISO execution error, the next step SHALL be skipped.

	2. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.4 Test case ISO7816_H_3

Purpose	READ BINARY command with SFI for EF.DG3 within an established PACE or BAC session, but without the terminal authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG3
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given READ BINARY APDU for EF.DG3 (SFI '03') to the eMRTD. Though PACE or BAC and the CA mechanisms have been performed, the access to the data group 3 SHALL be denied. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.5 Test case ISO7816_H_4

Purpose	READ BINARY command with SFI for EF.DG4 within an established PACE or BAC session, but without the terminal authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG4
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given READ BINARY APDU for EF.DG4 (SFI '04') to the eMRTD. Though PACE or BAC and the CA mechanisms have been performed, the access to the data group 4 SHALL be denied. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.6 Test case ISO7816_H_5

Purpose	READ BINARY command with odd instruction byte and SFI for EF.DG3 within an established PACE or BAC session, but without the terminal authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG3, OddIns
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given READ BINARY APDU for EF.DG3 (SFI '03') to the eMRTD. Though PACE or BAC and the CA mechanisms have been performed, the access to the data group 3 SHALL be denied. '0C B1 00 03 17 85 <L_{85}> <Cryptogram> 97 01 07 8E 08 <Checksum> 00' - The cryptogram contains the encrypted data object 54 with the encoded offset of 00 ('54 01 00').
Expected results	1. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.7 Test case ISO7816_H_6

Purpose	READ BINARY command with odd instruction byte and SFI for EF.DG4 within an established PACE or BAC session, but without the terminal authentication
Version	3.00

References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG4, OddIns
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given READ BINARY APDU for EF.DG4 (SFI '04') to the eMRTD. Though PACE or BAC and the CA mechanisms have been performed, the access to the data group 4 SHALL be denied. '0C B1 00 04 17 85 <L_{85}> <Cryptogram> 97 01 07 8E 08 <Checksum> 00' - The cryptogram contains the encrypted data object 54 with the encoded offset of 00 ('54 01 00').
Expected results	1. ISO checking or ISO execution error within a valid SM response
Post conditions	2. None

3.6.8 Test case ISO7816_H_7

Purpose	SELECT command for EF.CVCA without established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU
Preconditions	1. The "Open LDS Application" procedure SHALL **NOT** have been performed. 2. The Chip Authentication mechanism SHALL have been performed.
Test scenario	1. Select the ePassport application. '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the given SELECT APDU for EF.CVCA (<fid.EF.CVCA>) to the eMRTD. Since the "Open LDS Application" procedure has not been performed, the access to this file SHALL be denied. '00 A4 02 0C 02 <fid.EF.CVCA>' 3. Some chip implementations allow the selection of a protected file. In these cases, an additional READ BINARY SHOULD be used to verify that at least the READ BINARY command is prohibited. '00 B0 00 00 01'
Expected results	1. ISO checking or ISO Execution error or '90 00' as a plain response without Secure Messaging. If this step returns ISO checking or ISO Execution error, the next steps SHALL be skipped. 2. ISO checking or ISO Execution error or '90 00' as a plain response without Secure Messaging 3. ISO checking or ISO Execution error as a plain response without Secure Messaging
Post conditions	1. None

3.6.9 Test case ISO7816_H_8

Purpose	READ BINARY command with SFI for EF.CVCA without established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU
Preconditions	1. The SFI information from data group 14 TerminalAuthenticationInfo element must be used if present. Otherwise, the default value has to be used. If the TerminalAuthenticationInfo element specifies a file ID, but no short file ID, this test case is skipped.
Test scenario	1. Select the ePassport application. '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the given READ BINARY APDU for EF.CVCA (<sfi.EF.CVCA>) to the eMRTD. Since the "Open LDS Application" procedure has not been

	performed, the access to the EF.CVCA has to be denied. '00 B0 <sfi.EF.CVCA> 00 01'
Expected results	1. ISO checking or ISO Execution error or '90 00' as a plain response without Secure Messaging. If this step returns ISO checking or ISO Execution error, the next steps SHALL be skipped. 2. ISO checking or ISO Execution error as a plain response without Secure Messaging
Post conditions	1. None

3.6.10 Test case ISO7816_H_9

Purpose	READ BINARY command with odd instruction byte and with SFI for EF.CVCA without established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU, OddIns
Preconditions	1. The SFI information from data group 14 TerminalAuthenticationInfo element must be used if present. Otherwise, the default value has to be used. If the TerminalAuthenticationInfo element specifies a file ID, but no short file ID, this test case is skipped.
Test scenario	1. Select the ePassport application. '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the given READ BINARY APDU for EF.CVCA (<sfi.EF.CVCA>) to the eMRTD. Since the "Open ePassport Application" procedure has not been performed, the access to the EF.CVCA has to be denied. '00 B1 00 <sfi.EF.CVCA> 03 54 01 00 07'
Expected results	1. ISO checking or ISO Execution error or '90 00' as a plain response without Secure Messaging. If this step returns ISO checking or ISO Execution error, the next steps SHALL be skipped. 2. ISO checking or ISO Execution error as a plain response without Secure Messaging
Post conditions	1. None

3.6.11 Test case ISO7816_H_10

Purpose	READ BINARY command with odd instruction byte and with FID for EF.CVCA without established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU, OddIns
Preconditions	1. The fileID information from data group 14 TerminalAuthenticationInfo element must be used if present. Otherwise, the default value has to be used.
Test scenario	1. Select the ePassport application. '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the given READ BINARY APDU for EF.CVCA (<fid.EF.CVCA>) to the eMRTD. Since the "Open ePassport Application" procedure has not been performed, the access to the EF.CVCA has to be denied. '00 B1 <fid.EF.CVCA> 03 54 01 00 07'
Expected results	1. ISO checking or ISO Execution error or '90 00' as a plain response without Secure Messaging. If this step returns ISO checking or ISO Execution error, the next steps SHALL be skipped. 2. ISO checking or ISO Execution error as a plain response without Secure Messaging
Post conditions	1. None

3.6.12 Test case ISO7816_H_11

Purpose	READ BINARY command with odd instruction byte and FID for EF.DG3 within an established PACE or BAC session, but without the terminal authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG3, OddIns
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given READ BINARY APDU for EF.DG3 (FID '0103') to the eMRTD. Though "Open ePassport Application" procedure and the CA mechanisms have been performed, the access to the data group 3 SHALL be denied. '0C B1 01 03 17 85 <L_{85}> <Cryptogram> 97 01 07 8E 08 <Checksum> 00' - The cryptogram contains the encrypted data object 54 with the encoded offset of 00 ('54 01 00').
Expected results	1. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.13 Test case ISO7816_H_12

Purpose	READ BINARY command with odd instruction byte and FID for EF.DG4 within an established PACE or BAC session, but without the terminal authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-LDS1, DG4, OddIns
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The Chip Authentication mechanism SHALL have been performed, too.
Test scenario	1. Send the given READ BINARY APDU for EF.DG4 (FID '0104') to the eMRTD. Though "Open LDS Application" procedure and the CA mechanisms have been performed, the access to the data group 4 SHALL be denied. '0C B1 01 04 17 85 <L_{85}> <Cryptogram> 97 01 07 8E 08 <Checksum> 00' - The cryptogram contains the encrypted data object 54 with the encoded offset of 00 ('54 01 00').
Expected results	1. ISO checking or ISO execution error within a valid SM response
Post conditions	1. None

3.6.14 Test case ISO7816_H_13

Purpose	SELECT command for EF.CVCA with established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The fileID information from data group 14 TerminalAuthenticationInfo element SHALL be used if present. Otherwise, the default value has to be used.
Test scenario	1. Send the given SELECT APDU for EF.CVCA (<fid.EF.CVCA>) to the eMRTD. '0C A4 02 0C 15 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - The cryptogram contains the encrypted fileID of the EF.CVCA file (<fid.EF.CVCA>).

	2. The size of the EF_CVCA SHALL be 36 bytes. So try to read the entire EF.CVCA file with a single READ BINARY Command. '0C B0 00 00 0D 97 01 24 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid SM response 2. 36 bytes of content data and '90 00' in an SM response
Post conditions	1. None

3.6.15 Test case ISO7816_H_14

Purpose	READ BINARY command with SFI for EF.CVCA with established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The SFI information from data group 14 TerminalAuthenticationInfo element must be used if present. Otherwise, the default value has to be used. If the TerminalAuthenticationInfo element specifies a file ID, but no short file ID, this test case is skipped.
Test scenario	1. Send the given READ BINARY APDU for EF.CVCA (<sfi.EF.CVCA>) to the eMRTD. The size of the EF_CVCA SHALL be 36 bytes. So try to read the entire EF.CVCA file with a single READ BINARY Command. '0C B0 <sfi.EF.CVCA> 00 0D 97 01 24 8E 08 <Checksum> 00'
Expected results	1. 36 bytes of content data and '90 00' in an SM response
Post conditions	1. None

3.6.16 Test case ISO7816_H_15

Purpose	READ BINARY command with odd instruction byte and with SFI for EF.CVCA with established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU, OddIns
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The SFI information from data group 14 TerminalAuthenticationInfo element must be used if present. Otherwise, the default value has to be used. If the TerminalAuthenticationInfo element specifies a file ID, but no short file ID, this test case is skipped.
Test scenario	1. Send the given READ BINARY APDU for EF.CVCA (<sfi.EF.CVCA>) to the eMRTD. The size of the EF_CVCA SHALL be 36 bytes. So try to read the EF.CVCA file with a single READ BINARY Command. '0C B1 00 <sfi.EF.CVCA> 17 85 <L_{85}> <Cryptogram> 97 01 26 8E 08 <Checksum> 00' - The cryptogram contains the encrypted data object 54 with an encoded offset of 00 ('54 01 00').
Expected results	1. 38 bytes of data including the tag 53 and the BER encoded length. The Status must be '90 00'. The response must be protected by Secure Messaging.
Post conditions	1. None

3.6.17 Test case ISO7816_H_16

Purpose	READ BINARY command with odd instruction byte and with FID for EF.CVCA with established PACE or BAC session
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU, OddIns
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed.

	2. The SFI information from data group 14 TerminalAuthenticationInfo element must be used if present. Otherwise, the default value has to be used. If the TerminalAuthenticationInfo element specifies a file ID, but no short file ID, this test case is skipped.
Test scenario	1. Send the given READ BINARY APDU for EF.CVCA (<fid.EF.CVCA>) to the eMRTD. The size of the EF_CVCA SHALL be 36 bytes. So try to read the EF.CVCA file with a single READ BINARY Command. '0C B1 <fid.EF.CVCA> 17 85 <L_{85}> <Cryptogram> 97 01 26 8E 08 <Checksum> 00' - The cryptogram contains the encrypted data object 54 with an encoded offset of 00 ('54 01 00').
Expected results	1. 38 bytes of data including the tag 53 and the BER encoded length. The Status must be '90 00'. The response must be protected by Secure Messaging.
Post conditions	1. None

3.7 Unit Test ISO7816_I – Chip Authentication

3.7.1 Introduction

The chip authentication mechanism verifies that the chip is genuine and establishes secure messaging session keys. The terminal and the eMRTD generate a shared secret based on the public key data stored in LDS data group 14 file of the document. This secret is used to derive new session keys for the continued secure messaging session. The genuineness of the eMRTD chip is implicitly verified by its ability to perform Secure Messaging using the new session keys. The test cases specified in this unit verify the correct implementation of the "MSE:Set KAT" command and "MSE:Set AT" / "General Authenticate" command pair. According to [R1] Part 11 6.2.4, MSE: Set KAT SHALL NOT be used for any other algorithms than id-CA-DH-3DES-CBC-CBC and id-CA-ECDH-3DES-CBC-CBC, i.e. Secure Messaging is restricted to 3DES.
Data group 14 file conditionally contains a key reference identifier ([R1] Part 11 9.2.6). All tests in this unit SHALL be used with implicit or explicit key reference depending on presence of ambiguous ChipAuthenticationPublicKeyInfo, i.e more than one ChipAuthenticationPublicKeyInfo elements are present in LDS data group 14.
Data group 14 may contain more than one ChipAuthenticationInfo. In this case, all tests must be performed for each ChipAuthenticationInfo. The corresponding test case is only rated as a PASS if all test runs are completed successfully.

All test cases of this test unit which require the "Open LDS Application" procedure (see subclause 2.1) SHALL be performed twice (one test run with BAC and one with PACE) if the chip supports both protocols.
If not otherwise specified, if the chip supports Chip Authentication Mapping, PACE-CAM protocol SHALL NOT be used for PACE.
If the chip only supports one of these protocols (BAC or PACE), only one test run SHALL be performed with the supported protocol used in the "Open LDS Application" procedure.

3.7.2 Test Case ISO7816_I_1

Purpose	MSE:Set KAT command with correct ephemeral public key
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.

Test scenario	1. Send the given MSE APDU to the eMRTD.
	'0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>'
	- <Cryptogram> contains the following encrypted data objects
	`91 <L91> <ephemeral public key>`
	`84 <L84> <private key reference>`
	- The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present.
	2. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS SHALL be sent as SM-protected APDU using the new session keys.

Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the NEW session keys.

3.7.3 Test Case ISO7816_I_2

Purpose	MSE:Set KAT command with correct ephemeral public key, but afterwards the old session keys are used.
Version	2.11
References	[R1] Part 11 6.2

Profile	CA, CA-LDS1, CA-KAT
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Instead of using the new session keys, the session keys derived in step 1 of the test preconditions are used to send the Command APDU as defined in the ICS SM-protected APDU.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. ISO checking error. The chip SHALL delete the session keys derived in step 1 of the test preconditions and SHALL NOT accept any APDUs under Secure Messaging with these session keys. The SW must be returned as plain response without Secure Messaging.

3.7.4 Test Case ISO7816_I_3

Purpose	MSE:Set KAT command with invalid ephemeral public key (different key size)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - The ephemeral public key SHALL be generated with domain parameters specifying a different key size (e.g. for a 224 bits key in DG14 a 192 bits ephemeral key pair is created) - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify that the session keys derived in step 1 of the test preconditions can still be used, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test precondition.
Expected results	1. ISO checking error, or warning '63 00' within a valid SM response. Since there are invalid domain parameters used to generate the ephemeral key pair, the key agreement process SHALL always fail. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.5 Test Case ISO7816_I_4

Purpose	MSE:Set KAT command with a valid ephemeral public key, but without established PACE or BAC session
Version	2.11

References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT
Preconditions	1. The "Open LDS Application" procedure SHALL NOT have been performed. The content of ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL known to the test setup to generate an ephemeral key pair.
Test scenario	1. Select the LDS application. '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the given MSE APDU to the eMRTD. '00 22 41 A6 <Lc> 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> <Le>' - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 3. To verify that the chip does not activate the new session keys based on the key agreement, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys based on step 2.
Expected results	1. ISO checking error or '90 00' as a plain response without Secure Messaging. If this step returns ISO checking error, the next steps SHALL be skipped. 2. ISO checking error. The "Open LDS Application" procedure SHALL have been performed before the Chip Authentication can be done. The error code SHALL be returned as plain data without SM encoding. 3. ISO checking error. The error code SHALL be returned as plain data without SM encoding.

3.7.6 Test Case ISO7816_I_5

Purpose	MSE:Set KAT command with a valid ephemeral public key, but without SecureMessaging
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '00 22 41 A6 <Lc> 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> <Le>' - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify that the chip does not activate the new session keys based on the key agreement, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys. 3. To verify that the chip has deleted the session keys derived in step 1 of the test preconditions, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test preconditions.
Expected results	1. ISO checking error. The use of SecureMessaging SHALL be enforced by the chip. The error code SHALL be returned as plain data without SM encoding. 2. ISO checking error. The error code SHALL be returned as plain data without SM encoding. 3. ISO checking error. The error code SHALL be returned as plain data without SM encoding.

3.7.7 Test Case ISO7816_I_6

Purpose	MSE:Set KAT command with invalid data object tag for the ephemeral public key
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 93 <L_{93}> <ephemeral public key> 84 <L_{84}> <private key reference> - The data object for the ephemeral public key has an invalid tag 93. - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the preconditions.
Expected results	1. ISO checking error. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 2. '90 00' and a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.8 Test Case ISO7816_I_7

Purpose	MSE:Set KAT providing a (0,0) public key
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - The public key has to be coded as '04||x||y' where both x and y have a size according to the prime, but filled with '00'. - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test preconditions.
Expected results	1. ISO checking error or warning processing '63 00'. Note: Even if public key validation is not done, ECDH computation SHOULD fail with this input. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.9 Test Case ISO7816_I_8

Purpose	MSE:Set KAT test borderline cases for x- and y- coordinates (small x coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - Use an ephemeral public key with an x-coordinate requiring less than [$\log_{256}$ q] bytes to be represented. Pad with prepended zero bytes. (For details on q see [R5]) - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.10 Test Case ISO7816_I_9

Purpose	MSE:Set KAT test borderline cases for x- and y- coordinates (large x coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - Use an ephemeral public key with an x-coordinate having its most significant bit set to 1 - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.11 Test Case ISO7816_I_10

Purpose	MSE:Set KAT test borderline cases for x- and y- coordinates (small y coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - Use an ephemeral public key with an y-coordinate requiring less than [$\log_{256}$ q] bytes to be represented. Pad with prepended zero bytes. (For details on q see [R5]) - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.12 Test Case ISO7816_I_11

Purpose	MSE:Set KAT test borderline cases for x- and y- coordinates (large y coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - Use an ephemeral public key with an y-coordinate having its most significant bit set to 1 - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.13 Test Case ISO7816_I_12

Purpose	MSE:Set KAT command with an incorrect private key reference
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-KEYREF
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <invalid private key reference> - A private key reference SHALL be included in the APDU. This key reference SHALL be used as defined in the ICS. 2. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the preconditions.
Expected results	1. ISO checking error or warning processing '63 00'. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 2. '90 00' and a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.14 Test Case ISO7816_I_13

Purpose	Check the Chip authentication failure (using DH) – wrong value (value strictly bigger than the Prime)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-DH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - Use an ephemeral public key with a wrong value (value strictly bigger than the Prime) ephemeral public key = prime p + 1 - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test preconditions.
Expected results	1. ISO checking error or warning processing '63 00'. The SW SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.15 Test Case ISO7816_I_14

Purpose	Check the Chip authentication failure (using ECDH) – wrong point (value does not belong to the curve)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-KAT, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> < private key reference> - Use an ephemeral public key with a wrong point (value does not belong to the curve) - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the preconditions.
Expected results	1. ISO checking error or warning processing '63 00'. The SW SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.16 Test Case ISO7816_I_15

Purpose	MSE:Set KAT command with correct ephemeral public key after PACE-CAM
Version	3.00
References	[R1] Part 11 6.2
Profile	PACE, PACE-CAM, CA, CA-LDS1, CA-KAT
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed with PACE-CAM. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE APDU to the eMRTD. '0C 22 41 A6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 91 <L_{91}> <ephemeral public key> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS SHALL be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the NEW session keys.

3.7.17 Test Case ISO7816_I_16

Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key

Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> 3. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test precondition. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the NEW session keys.

3.7.18 Test Case ISO7816_I_17

Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key, but afterward the old session keys are used.
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> 3. Instead of using the new session keys, the session keys derived in step 1 of the test preconditions are used to send the Command APDU as defined in the ICS as SM-protected APDU.

Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. ISO checking error. The chip SHALL deletes the session keys which were derived in step 1 of the test preconditions and SHALL NOT accept any APDUs under Secure Messaging with these session keys. The error must be a returned as plain response without Secure Messaging.

3.7.19 Test Case ISO7816_I_18

Purpose	MSE:Set AT / General Authenticate commands with invalid ephemeral public key (different key size)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> - The ephemeral public key SHALL be generated with domain parameters specifying a different key size (e.g. for a 224 bits key in DG14 a 192 bit ephemeral key pair is created) 3. To verify that the session keys derived in step 1 of the test preconditions can still be used, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test preconditions.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. ISO checking error, or warning SW '63 00'. If chip returns SW '63 00', response data field MAY contain '7C 00'. If chip returns an ISO checking error SW, response data field SHALL be absent. Since there are invalid domain parameters used to generate the ephemeral key pair, the key agreement process SHALL always fail. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.20 Test Case ISO7816_I_19

Purpose	MSE:Set AT / General Authenticate commands with a valid ephemeral public key, but without established PACE or BAC session
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA
Preconditions	1. The "Open LDS Application" procedure SHALL NOT have been

	performed. The content of ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL known to the test setup to generate an ephemeral key pair.
Test scenario	1. Select the LDS application. '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the given MSE:Set AT APDU to the eMRTD. '00 22 41 A4 <Lc> 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> 00' - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 3. Send the given General Authenticate APDU to the eMRTD. '00 86 00 00 <Lc> 7C <L7C> 80 <L80> <ephemeral public key> <Le>' 4. To verify that the chip does not activate the new session keys based on the key agreement, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the expected session keys based on step 3.
Expected results	1. ISO checking error or '90 00' as a plain response without Secure Messaging. If this step returns ISO checking error, the next steps SHALL be skipped. 2. ISO checking error or '90 00' as a plain response without Secure Messaging. Note that some chip OS accept the selection of an unavailable private key and return an error only when the private key is used for the selected purpose. 3. ISO checking error or '90 00' as a plain response without Secure Messaging. 4. ISO checking error. The error code SHALL be returned as plain data without SM encoding.

3.7.21 Test Case ISO7816_I_20

Purpose	MSE:Set AT / General Authenticate commands with a valid ephemeral public key, but without SecureMessaging
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD (without Secure Messaging). '00 22 41 A4 <Lc> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference> 00' - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD (without Secure Messaging). '00 86 00 00 <Lc> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <Le>' 3. To verify that the chip has deleted the session keys derived in step 1 of the test preconditions, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test precondition.
Expected results	1. ISO checking error or '90 00'. In case of an error code, it SHALL be returned as plain data without SM encoding. 2. ISO checking error. The error code SHALL be returned as plain data without SM encoding. 3. ISO checking error. The error code SHALL be returned as plain data without SM encoding.

3.7.22 Test Case ISO7816_I_21

Purpose	MSE:Set AT / General Authenticate commands with invalid data object tag for the ephemeral public key
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 81 <L_{81}> <ephemeral public key> - The data object for the ephemeral public key has an invalid tag 81. 3. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step1 of the test precondition.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. ISO checking error. Response data field SHALL be absent. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.23 Test Case ISO7816_I_22

Purpose	MSE:Set AT / General Authenticate commands, providing a (0,0) public key to General Authenticate
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>'

	- <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> - The public key has to be coded as '04||x||y' where both x and y have a size according to the prime, but filled with '00'. 3. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1of the test precondition.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. ISO checking error or warning SW '63 00'. If chip returns SW '63 00', response data field MAY contain '7C 00'. If chip returns an ISO checking error SW, response data field SHALL be absent. Note: Even if public key validation is not done, DH computation SHOULD fail with this input. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.24 Test Case ISO7816_I_23

Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y- coordinates (small x coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> - Use an ephemeral public key with an x-coordinate requiring less than [$\log_{256}$ q] bytes to be represented. Pad with prepended zero bytes. (For details on q see [R5]) 3. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.25 Test Case ISO7816_I_24

Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y- coordinates (large x coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> - Use an ephemeral public key with an x-coordinate having its most significant bit set to 1 3. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.26 Test Case ISO7816_I_25

Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y- coordinates (small y coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>'

	- <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> - Use an ephemeral public key with an y-coordinate requiring less than [$\log_{256}$ q] bytes to be represented. Pad with zero bytes. (For details on q see [R5]) 3. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.27 Test Case ISO7816_I_26

Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y- coordinates (large y coordinate)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> - Use an ephemeral public key with an y-coordinate having its highest bit set to 1 3. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. '90 00' and a valid Secure Messaging response. The returned data SHALL be encoded with the new session keys.

3.7.28 Test Case ISO7816_I_27

Purpose	MSE:Set AT command with an incorrect private key reference
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-KEYREF

Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <invalid private key reference> - A private key reference SHALL be included in the APDU. This key reference SHALL be different from the one potentially specified in the ChipAuthenticationPublicKeyInfo structure stored in LDS data group 14 (see ICS). 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> 3. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test precondition.
Expected results	1. ISO checking error or warning processing '63 00' or '90 00'. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 2. ISO checking error or warning processing '63 00'. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. This step is performed only if '90 00' has been returned in step 1 3. '90 00' and a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.29 Test Case ISO7816_I_28

Purpose	Check the Chip authentication failure (using DH) – wrong value (value strictly bigger than the Prime)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-DH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L7C> 80 <L80> <ephemeral public key> - Use an ephemeral public key with a wrong value (value strictly bigger than the Prime) ephemeral public key = prime p + 1

	3. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the precondition.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. ISO checking error or warning SW '63 00'. If chip returns SW '63 00', response data field MAY contain '7C 00'. If chip returns an ISO checking error SW, response data field SHALL be absent. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 3. '90 00' and a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.30 Test Case ISO7816_I_29

Purpose	Check the Chip authentication failure (using ECDH) – wrong point (value does not belong to the curve)
Version	2.11
References	[R1] Part 11 6.2
Profile	CA, CA-LDS1, CA-ATGA, CA-ECDH
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L80> <cryptographic mechanism reference> 84 <L84> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L7C> 80 <L80> <ephemeral public key> - Use an ephemeral public key with a wrong point (value does not belong to the curve) 3. To verify that the session keys derived in step 1 of the test preconditions are still valid, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the session keys derived in step 1 of the test precondition.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 2. ISO checking error or warning SW '63 00'. If chip returns SW '63 00', response data field MAY contain '7C 00'. If chip returns an ISO checking error SW, response data field SHALL be absent. The SW SHALL be encoded in a Secure Messaging response using the session keys derived in step 1 of the test preconditions. 3. '90 00' and a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions.

3.7.31 Test Case ISO7816_I_30

Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key after PACE-CAM
Version	3.00
References	[R1] Part 11 6.2

Profile	PACE, PACE-CAM, CA, CA-LDS1, CA-ATGA
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. 2. The ChipAuthenticationPublicKeyInfo stored in LDS data group 14 SHALL have been read to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eMRTD. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <cryptographic mechanism reference> 84 <L_{84}> <private key reference> - The private key reference SHALL be included in the APDU if and only if ChipAuthenticationPublicKeyInfo is ambiguous, i.e more than one ChipAuthenticationPublicKeyInfo elements are present. 2. Send the given General Authenticate APDU to the eMRTD. '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' - <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> 3. To verify the chip's ability to continue the Secure Messaging with the new session keys, the Command APDU as defined in the ICS must be sent as SM-protected APDU using the new session keys.
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test precondition. 2. '7C 00 90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the session keys derived in step 1 of the test preconditions. 3. '90 00' in a valid Secure Messaging response. The returned data SHALL be encoded with the NEW session keys.

3.8 Unit Test ISO7816_J – Certificate verification

3.8.1 Introduction

During the Terminal Authentication process the certificate chain from the trust point returned by the PACE protocol or stored in the chip's EF.CVCA down to the terminal certificate is verified. This is done by an alternating sequence of MSE: Set DST and Verify Certificate commands. This unit covers all certificate verification test cases which do NOT update the chips persistent memory. This means that all tests in this unit can be repeated with the same set of certificates.
The certificates used in this Unit Test are defined in Certificate Set 1 of this document.

3.8.2 Preconditions

The preconditions depend on the applicable profiles defined in clause 2.2.

3.8.2.1 Preconditions for TA-EU

Apply preconditions defined below:

1. The "Open LDS Application" procedure (see subclause 2.1) SHALL have been performed[1].
2. The Chip Authentication mechanism SHALL have been performed as well.
3. The Certification Authority Reference SHALL has been read from the EF.CVCA file (Primary trust point).
4. All APDUs are sent as valid Secure Messaging APDUs.
5. All response data SHALL be SM protected.

3.8.2.2 Preconditions for TA-LDS1

To be defined.

3.8.2.3 Preconditions for TA-MF

To be defined.

3.8.3 Test case ISO7816_J_1

Purpose	Positive test with a valid chain of CV certificates.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>'

[1] PACE must be used if supported

	<Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. '90 00' in a valid Secure Messaging response. 3. '90 00' in a valid Secure Messaging response. 4. '90 00' in a valid Secure Messaging response.
Post conditions	None

3.8.4 Test case ISO7816_J_2

Purpose	Test with an invalid Certification Authority Reference.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <BAD certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. but the last character to create an invalid reference. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.5 Test case ISO7816_J_3

Purpose	Test with an invalid certificate signature.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <BAD certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.6 Test case ISO7816_J_4

Purpose	Test with a missing certificate signature.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1.

	'0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> The certificate signature object is omitted. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.7 Test case ISO7816_J_5

Purpose	Test with a missing certificate body.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 5F 37 <L_{5F37}> <certificate signature> The certificate body object is omitted. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.8 Test case ISO7816_J_6

Purpose	Test a DV certificate with a missing Holder Authorization.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 5F 37 <L_{5F37}> <certificate signature> The certificate does not contain a certificate holder authorization certificate body object is omitted. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified

	successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.9 Test case ISO7816_J _7

Purpose	Test a DV certificate with a missing effective date.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_2. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate does not have a certificate effective date tag. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.10 Test case ISO7816_J_8

Purpose	Test a DV certificate with a missing expiration date.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate does not have a certificate expiration date tag. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 5. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.11 Test case ISO7816_J_9

Purpose	Test a DV certificate with an incorrect encoded effective date. (bad BCD coding)
Version	3.00
References	[R1] Part 11 7.1 [R1] Part 12 7.2.3.1.3
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_4. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate contains a badly encoded BCD effective date.

	3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.12 Test case ISO7816_J_10

Purpose	Test a DV certificate with an incorrect encoded expiration date. (bad BCD coding)
Version	3.00
References	[R1] Part 11 7.1 [R1] Part 12 7.2.3.1.3
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_5. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate contains a badly encoded BCD expiration date. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects

	7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid Secure Messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.13 Test case ISO7816_J_11

Purpose	Test the "Current Date" update mechanism with a new foreign T certificate.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified DV_CERT_1_11. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-certificate is marked as a foreign DV-certificate. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate has an advanced effective date. Since the DV certificate was marked as a foreign one, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 5. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08

	<Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 6. Send the appropriate DV-Certificate as specified in DV_CERT_1_11. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-certificate is marked as a foreign DV-certificate. 7. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 6 has to be used. 8. Send the appropriate T-Certificate as specified in T_CERT_1_2. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the T-Certificate used in step 4.
Expected results	1. '90 00' in a valid Secure Messaging response. 2. '90 00' in a valid Secure Messaging response. 3. '90 00' in a valid Secure Messaging response. 4. '90 00' in a valid Secure Messaging response. 5. '90 00' in a valid Secure Messaging response. 6. '90 00' in a valid Secure Messaging response. 7. '90 00' in a valid Secure Messaging response. 8. '90 00' in a valid Secure Messaging response. This certificate SHALL still be accepted since the chip SHALL NOT change the current date based on the foreign T certificate
Post conditions	None

3.8.14 Test case ISO7816_J_12

Purpose	Test with a valid chain of CV certificates but without using SecureMessaging.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '00 22 81 B6 <Lc> 83 <certificate authority reference>' The Certification Authority Reference SHALL be used as read from the EF.CVCA file. The APDU is send in plain without Secure Messaging. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body>

	5F 37 <L_{5F37}> <certificate signature> The APDU is send as a valid secure messaging APDU. After step 2, the eMRTD token is reset and the preconditions of this test case are reestablished 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. The APDU is send as a valid secure messaging APDU 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '00 2A 00 BE <Lc> 7F 4E <L7F4E> <body> 5F 37 <L5F37> <signature>' 5. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 4 has to be used. The APDU is send as a valid secure messaging APDU.
Expected results	1. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning. A chip may permit the use of an unprotected MSE APDU, however, the SM channel SHALL be closed as soon as an unprotected APDU is send. Therefore, the response Must be sent without SM encoding. 2. ISO_checking_error or ISO_execution_error or ISO_warning. Since the secure messaging channel SHALL have been closed in Step 1, the chip SHALL return an error without secure messaging encoding here. 3. '90 00' in a valid secure messaging response 4. ISO_checking_error or ISO_execution_error or ISO_warning. The secure messaging channel SHALL be closed as soon as an unprotected APDU is send. The error code SHALL be returned as plain data without secure messaging encoding. 5. ISO_checking_error or ISO_execution_error or ISO_warning. Since the secure messaging channel SHALL have been closed in Step 4, the chip SHALL return an error without secure messaging encoding here.
Post conditions	None

3.8.15 Test case ISO7816_J_13

Purpose	Test with an invalid certificate body tag.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4F <L_{7F4F}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>

	The certificate body tag has been changed to '7F 4F'. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.16 Test case ISO7816_J_14

Purpose	Test with an invalid certificate signature tag.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 38 <L_{5F38}> <certificate signature> The certificate signature tag has been changed to '5F 38'. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects

	7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.17 Test case ISO7816_J_15

Purpose	Test a DV certificate with an incorrect Gregorian effective date.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-DATE
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_6. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate contains an invalid Gregorian effective date. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-

	Certificate verification.
Post conditions	None

3.8.18 Test case ISO7816_J_16

Purpose	Test a DV certificate with an incorrect Gregorian expiration date.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-DATE
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_7. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate contains an invalid Gregorian expiration date. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.19 Test case ISO7816_J_17

Purpose	Test a DV certificate with an expiration date BEFORE the effective date.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	<Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_8. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate contains an expiration date BEFORE the effective date. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.20 Test case ISO7816_J_18

Purpose	Test correct removal of temporary keys.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Reset the chip, perform the “Open ePassport Application” procedure and the Chip Authentication. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. The tempory key of the DV certificate SHALL have been deleted during the reset. Therefore it SHALL NOT be possible to verify the IS certificate based on this key.
Post conditions	None

3.8.21 Test case ISO7816_J_19

Purpose	Test a DV certificate with invalid OID in the Certificate Holder Authorization element.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_9. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate has an invalid OID in the Certificate holder Authorization element. Note: If the chip supports further OIDs in addition to the ones specified in R2], this SHALL be stated in the ICS. For this test an OID SHALL be used which is NOT supported by the chip. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body>

	5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.22 Test case ISO7816_J_20

Purpose	Test a DV certificate with invalid OID in the Public Key element
Version	1.0
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_10. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate has an invalid OID in thePublic Key element. 3. Send the given MSE: Set DST APDU to the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid Secure Messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.

Post conditions	None

3.8.23 Test case ISO7816_J_21

Purpose	Test the CVCA root key selection with a wrong name (CAR) - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with a wrong CAR. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted wrong CVCA key Name 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_12. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate is issued by the CVCA whose selection SHOULD have failed. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with a correct CVCA key name (CAR). '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. A chip may permit the selection of an unknown key. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.24 Test case ISO7816_J_22

Purpose	Test a DV certificate with a wrong certificate body tag - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_13. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4F <L_{7F4F}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The tag of the certificate body is wrong. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. A chip may permit the selection of an unknown key. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.25 Test case ISO7816_J_23

Purpose	Test a DV certificate with a wrong certificate signature tag - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_14. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 38 <L_{5F38}> <certificate signature> The tag of the certificate signature is wrong. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date.

	Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.26 Test case ISO7816_J_24

Purpose	Test a DV certificate with a wrong certificate body length - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified DV_CERT_1_15. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> + 1 <certificate body> 5F 37 <L_{5F37}> <certificate signature> The length of the certificate body is unconsistent. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects

	7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.27 Test case ISO7816_J_25

Purpose	Test a DV certificate with a wrong certificate signature length - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_16. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> + 1 <certificate signature> The length of the certificate signature is unconsistent. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.28 Test case ISO7816_J_26

Purpose	Test a DV certificate with a wrong certificate signature (Last byte increased by 1) - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_17. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature + 1> The certificate signature is wrong. It is obtained by increasing a correct signature by one. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.29 Test case ISO7816_J_27

Purpose	Test a DV certificate with a wrong certificate signature (Dropping last byte of the signature) - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2.

	2. Send the appropriate DV-Certificate as specified in DV_CERT_1_18. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate signature is wrong. It is obtained by dropping the last byte of the certificate signature (the length of the DO remains consistent). This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.30 Test case ISO7816_J_28

Purpose	Test a DV certificate with a wrong certificate signature (Signature greater than the modulus) - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_19. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate signature is wrong. It is obtained by It is obtained by setting the signature to a value greater than the modulus. The length of the signature SHALL match the length of the modulus. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case

	before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.31 Test case ISO7816_J_29

Purpose	Test a DV certificate with a wrong certificate signature (r = 0) - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_20. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate signature is wrong. It is obtained by filling the 'r' part of the signature with '00'. The length of 'r' is still matches the size of the prime. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body>

	5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.32 Test case ISO7816_J_30

Purpose	Test a DV certificate with a wrong certificate signature (s = 0) - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_21. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The certificate signature is wrong. It is obtained by filling the 's' part of the signature with '00'. The length of 's' is still matches the size of the prime. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.33 Test case ISO7816_J_31

Purpose	Test a DV certificate without selecting any root key - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the appropriate DV-Certificate as specified in DV_CERT_1_12. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> As no current key is selected, the certificate verification SHOULD fail. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 2. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 3. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.34 Test case ISO7816_J_32

Purpose	Test a DV certificate while the Public Key DO has a wrong OID field - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_22. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The Public Key DO in the certificate body contains an uncorrect OID that

	does not indicate id-TA (0.4.0.127.0.7.2.2.3.x.y). This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.35 Test case ISO7816_J_33

Purpose	Test a DV certificate while the Public Key DO has no OID field - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_23. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The Public Key DO in the certificate body does not contain an OID field. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1.

	'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.36 Test case ISO7816_J_34

Purpose	Test a DV certificate while the Public Key DO has no Public point field - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_24. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The Public Key DO in the certificate body does not contain any EC Public point field. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.

Post conditions	None

3.8.37 Test case ISO7816_J_35

Purpose	Test a DV certificate while the Public Key DO has no Modulus field - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_25. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The Public Key DO in the certificate body does not contain any RSA Modulus field. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.38 Test case ISO7816_J_36

Purpose	Test a DV certificate while the Public Key DO has no public exponent field - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_26 '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The Public Key DO in the certificate body does not contain any RSA public exponent field. This certificate has an advanced effective date. Since the DV certificate failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.39 Test case ISO7816_J_37

Purpose	Test a DV certificate while the Public Key DO contains an unknown DO - Current date not updated
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to initiate the certificate verification '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_27. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The Public Key DO in the certificate body contains an unknown DO (tag '77'). This certificate has an advanced effective date. Since the DV certificate

	failed, the chip SHALL NOT update the current date. Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the MSE: Set DST APDU to initiate the certificate verification to the eMRTD token with the CAR of the CVCA. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted CVCA key Name (CAR) The Certification Authority Reference SHALL be used as defined in 3.8.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response.
Post conditions	None

3.8.40 Test case ISO7816_J_38

Purpose	Test the transition CVCA ⇨ IS key
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate T-Certificate as specified in T_CERT_1_3. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.41 Test case ISO7816_J_39

Purpose	Test the transition CVCA ⇨ domestic DV ⇨ CVCA
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	<Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate CVCA-Certificate as specified in LINK_CERT_1_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.42 Test case ISO7816_J_40

Purpose	Test the transition CVCA ⇨ foreign DV ⇨ CVCA
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_11. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate CVCA-Certificate as specified in LINK_CERT_1_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08

	<Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.43 Test case ISO7816_J_41

Purpose	Test the transition CVCA ⇨ domestic DV ⇨ domestic DV
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1_28. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.44 Test case ISO7816_J_42

Purpose	Test the transition CVCA ⇨ domestic DV ⇨ foreign DV
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token

	'0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified in DV_CERT_1_29. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.45 Test case ISO7816_J_43

Purpose	Test the transition CVCA ⇨ foreign DV ⇨ domestic DV
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_11. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used.

	4. Send the appropriate DV-Certificate as specified in DV_CERT_1_28. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.46 Test case ISO7816_J_44

Purpose	Test the transition CVCA ⇨ foreign DV ⇨ foreign DV
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_11. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified DV_CERT_1_29. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.47 Test case ISO7816_J_45

Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ foreign DV
Version	3.00
References	[R1] Part 11 7.1
Profile	TA

Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the appropriate DV-Certificate as specified in DV_CERT_1_30. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose 6. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.48 Test case ISO7816_J_46

Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ domestic DV
Version	3.00
References	[R1] Part 11 7.1
Profile	TA

Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the appropriate DV-Certificate as specified in DV_CERT_1_31. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose 6. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.49 Test case ISO7816_J_47

Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ IS
Version	3.00
References	[R1] Part 11 7.1
Profile	TA

Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the appropriate T-Certificate as specified in T_CERT_1_4. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose 6. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.8.50 Test case ISO7816_J_48

Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ CVCA
Version	3.00
References	[R1] Part 11 7.1
Profile	TA

Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the appropriate CVCA-Certificate as specified in LINK_CERT_1_2. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose 6. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response
Post conditions	None

3.8.51 Test case ISO7816_J_49

Purpose	Test a DV certificate with a wrong Public Key (shorter key length).
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token

	'0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1_32. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The key length of this certificate is different to the CVCA public key. 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DVCA-Certificate sent. 4. Send the appropriate T-Certificate as specified in T_CERT_1_5. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 3. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. Since the DV certificate was not verified successfully, it SHALL NOT be possible to use it as the trust point for the T-Certificate verification.
Post conditions	None

3.8.52 Test case ISO7816_J_50

Purpose	Test a IS certificate with a wrong Public Key (shorter key length).
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.8.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD token '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.8.2. 2. Send the appropriate CA-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD token. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08

	<Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DVCA-Certificate sent. 4. Send the appropriate T-Certificate as specified in T_CERT_1_6. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The key length of this certificate is different to the CVCA and DV certificates public key.
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	None

3.9 Unit Test ISO7816_K – Terminal Authentication

3.9.1 Introduction

This unit tests the terminal authentication version 1. In this step, the terminal proves the possession of the private key which belongs to its certificate.
All test cases of this test unit which require the Open LDS Application procedure (see subclause 2.1) SHALL be performed for each access control mechanism supported by the chip. For instance, test cases SHALL be performed twice if the chip supports both protocols BAC and PACE (one test run with BAC and one with PACE). The corresponding test case is only rated as a PASS if all test cases are completed successfully.
If not otherwise specified, if the chip supports Chip Authentication Mapping, PACE-CAM protocol SHALL NOT be used for PACE during Open LDS Application procedure.
The certificates used in this Unit Test are defined in Certificate Set 1 of this document.

3.9.2 Preconditions

The preconditions depend on the applicable profiles defined in clause 2.2.

3.9.2.1 Preconditions for TA-EU

Apply preconditions defined below:

1. The Open LDS Application procedure (see subclause 2.1) SHALL have been performed.

Note: If BAC protocol is used, the chip's Document number as contain in the MRZ including the check digit ($ID_{IC)}$) SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command.
If PACE is used, the chip's ephemeral PACE public key SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. $ID_{IC} = Comp(PK_{DH,IC})$.

2. The Chip Authentication mechanism SHALL have been performed as well.
3. The Certification Authority Reference SHALL has been read from the EF.CVCA file (Primary trust point).
4. All APDUs are sent as valid Secure Messaging APDUs.
5. All response data SHALL be SM protected.

3.9.2.2 Preconditions for TA-LDS1

To be defined.

3.9.2.3 Preconditions for TA-MF

To be defined.

3.9.3 MSE: Set AT cryptogram

For Terminal Authentication version 1, the cryptogram used in MSE: Set AT command contains the following encrypted data objects:

1. 83 <L_{83}> <Certification Holder Reference>

3.9.4 Test case ISO7816_K_1

Purpose	Positive test with a valid terminal authentication process
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2.

	2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Check the Terminal Authentication protocol of the PublicKey data object of contained in DV certificate. 8. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2 <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '90 00' in a valid SM response 2. '90 00' in a valid SM response 3. '90 00' in a valid SM response 4. '90 00' in a valid SM response 5. '90 00' in a valid SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. The OID of the Public key data object is one of the following: - id-TA-RSA-PSS-SHA-256: (OID: 0.4.0.127.0.7.2.2.2.1.4) - id-TA-RSA-PSS-SHA-512: (OID: 0.4.0.127.0.7.2.2.2.1.6) - id-TA-ECDSA-SHA-224: (OID: 0.4.0.127.0.7.2.2.2.2.2) - id-TA-ECDSA-SHA-256: (OID: 0.4.0.127.0.7.2.2.2.2.3) - id-TA-ECDSA-SHA-384: (OID: 0.4.0.127.0.7.2.2.2.2.4) - id-TA-ECDSA-SHA-512: (OID: 0.4.0.127.0.7.2.2.2.2.5) Algotrithm RSA PKCS v1.5 and SHA-1 are not authorized. 8. '90 00' in a valid SM response
Post conditions	1. None

3.9.5 Test case ISO7816_K_2

Purpose	Test with an invalid certificate reference for the MSE:Set AT command
Version	3.00
References	[R1] Part 11 7.1
Profile	TA

Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. To generate an invalid certification holder reference, the last character of the holder reference stored inside the T-Certificate sent in step 4 is changed. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2 <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '90 00' in a valid SM response 2. '90 00' in a valid SM response 3. '90 00' in a valid SM response 4. '90 00' in a valid SM response 5. '90 00' or ISO checking error or ISO execution error in a valid SM response. Note that some chip OS accept the selection of an unavailable public key and return an error only when the public key is used for the selected purpose. 6. '<Eight bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response 7. ISO checking error or ISO execution error or '6300' in an SM response
Post conditions	1. None

3.9.6 Test case ISO7816_K_3

Purpose	Test with a terminal authentication process without secure messaging
Version	3.00

References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3 The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '00 82 00 00 <Lc> <Terminal generated signature>' The terminal generated signature SHALL be computed with ID_{IC} as defined in 3.9.2, but the APDU is sent in plain without SM encoding The signature is created with the private key of IS_KEY_01.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. ISO checking error or ISO execution error as a plain response (without Secure Messaging)
Post conditions	1. None

3.9.7 Test case ISO7816_K_4

Purpose	Test that the effective access rights in a DV-Certificate are ignored, i.e. sending a terminal certificate is skipped during TA and an error is expected
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2

Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3 The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 5. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2 <Cryptogram> contains the encrypted terminal generated signature created with the private key of DV_KEY_01.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' or ISO checking error or ISO execution error in an SM response 4. '<Eight bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response 5. ISO checking error or ISO execution error or '6300' in an SM response
Post conditions	1. None

3.9.8 Test case ISO7816_K_5

Purpose	Test that the effective access rights in a CVCA-Certificate are ignored, i.e. sending any certificate is skipped during TA and an error is expected
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Holder Reference > The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 3. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2 <Cryptogram> contains the encrypted terminal generated signature created with the private key of CVCA_KEY_00.
Expected results	1. '90 00' or ISO checking error or ISO execution error in an SM response

	2. '<Eight bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response 3. ISO checking error or ISO execution error or '6300' in an SM response
Post conditions	1. None

3.9.9 Test case ISO7816_K_7

Purpose	Terminal authentication process with two Get Challenge commands (Using the first challenge)
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given a second Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' If the chip returns an ISO checking error or ISO execution error for this second Get Challenge, skip the remaining step 8. 8. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The signature is based on the first challenge received in step 6
Expected results	1. '90 00' in an SM response

	2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '<Eight bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response 8. Skipped or ISO checking error or ISO execution error or '63 00' in an SM response
Post conditions	1. None

3.9.10 Test case ISO7816_K_8

Purpose	Terminal authentication process with short challenge
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 07 8E 08 <Checksum> 00' 7. If the chip returns a short challenge (only 7 bytes) then send the given external authenticate command to the eMRTD, otherwise skip this step. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.

	The signature is based on the short challenge received in step 6
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Seven bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response 7. Skipped, ISO checking error or ISO execution error or warning processing '63 00' in an SM response
Post conditions	1. None

3.9.11 Test case ISO7816_K_9

Purpose	Check the Terminal authentication – No Get Challenge Performed
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The wrong signature is calculated without any challenge. 7. To verify that access has NOT been granted after Terminal Authentication, an arbitrary SM APDU is sent to the chip. The Command APDU as defined in the ICS SHALL be sent.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response

	4. '90 00' in an SM response 5. '90 00' in an SM response 6. ISO checking error or ISO execution error or warning processing '63 00' in an SM response. 7. ISO checking error or ISO execution error in an SM response
Post conditions	1. None

3.9.12 Test case ISO7816_K_10

Purpose	Check the Terminal authentication – No authentication key selection performed
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' If the Get Challenge command returns an ISO checking error or ISO execution error, skip the remaining steps 6 and 7. 6. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The signature is based on the challenge received in step 5. 7. To verify that access has NOT been granted after Terminal Authentication, an arbitrary SM APDU is sent to the chip. The Command APDU as defined in the ICS SHALL be sent.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '<Eight bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response 6. Skipped, or ISO checking error or ISO execution error or warning processing '63 00' in an SM response 7. Skipped, or ISO checking error or ISO execution error in an SM response

Post conditions	1. None

3.9.13 Test case ISO7816_K_11

Purpose	Check the Terminal authentication – Wrong structure in the MSE Set AT command
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. For <Certification Holder Reference> tag, use '84' instead of '83' The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' If Get Challenge command returns an ISO checking error or ISO execution error, skip the remaining steps 7 and 8. 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The signature is based on the challenge received in step 6. 8. To verify that access has NOT been granted after Terminal Authentication, an arbitrary SM APDU is sent to the chip. The Command APDU as defined in the ICS SHALL be sent.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. ISO checking error or ISO execution error in an SM response 6. '<Eight bytes of random data> 90 00' or ISO checking error or ISO execution error in an SM response

	7. Skipped or ISO checking error or ISO execution error or warning processing '63 00' in an SM response 8. Skipped or ISO checking error or ISO execution error in an SM response
Post conditions	1. None

3.9.14 Test case ISO7816_K_12

Purpose	Check the Terminal authentication – Reset of the access rights in case of Application reset
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.9.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.9.2. 2. Send the appropriate DV-Certificate as specified DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted data objects as defined in 3.9.3. The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The signature is based on the challenge received in step 6. 8. Reset the chip by switching off the field and switching it on again Perform the first step of the preconditions defined in 3.9.2 To verify that access has NOT been granted after chip's reset, the Command APDU as defined in the ICS SHALL be sent.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response

	7. '90 00' in an SM response 8. ISO checking error or ISO execution error in an SM response
Post conditions	1. None

3.9.15 Test case ISO7816_K_13

Purpose	Check the Terminal Authentication – Passive and optional Active Authentication between Chip Authentication and Terminal authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. Note If BAC protocol is used, the chip's Document number as contain in the MRZ including the check digit SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. If PACE is used, the chip's ephemeral PACE public key SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. $ID_{IC} = Comp(PK_{DH,IC})$ 2. The Chip Authentication mechanism SHALL have been performed as well. 3. The Passive Authentication SHALL have been performed after CA. 4. If Active Authentication is supported, Active Authentication Public Key Info SHALL have been read and verified and Active Authentication SHALL have been performed after PA. 5. The Certification Authority Reference SHALL has been read from the EF.CVCA file (Primary trust point). 6. All APDUs are sent as valid Secure Messaging APDUs. 7. All response data SHALL be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1 '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83<L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'

	7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The signature is based on the challenge received in step 6.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response
Post conditions	1. None

3.9.16 Test case ISO7816_K_14

Purpose	Negative Test of Terminal Authentication with invalid PACE binding
Version	3.00
References	[R1] Part 11 7.1
Profile	TA-LDS1, PACE
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed. Note PACE SHALL be used, the chip's ephemeral PACE public key SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. $ID_{IC} = Comp(PK_{DH,IC})$ 2. The Chip Authentication mechanism SHALL have been performed as well. 3. The Certification Authority Reference SHALL has been read from the EF.CVCA file or PACE protocol (Primary trust point). 4. All APDUs are sent as valid Secure Messaging APDUs. 5. All response data SHALL be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>

	5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. Generate an invalid signature, e.g. modify last byte of the signature by adding 0x01. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. ISO checking error or ISO execution error or SW '63 00' in an SM response
Post conditions	1. None

3.9.17 Test case ISO7816_K_15

Purpose	Positive test with a valid terminal authentication process after PACE-CAM
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, PACE, PACE-CAM
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed with PACE-CAM. Note PACE SHALL be used, the chip's ephemeral PACE public key SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. $ID_{IC} = Comp(PK_{DH,IC})$ 2. The Chip Authentication mechanism SHALL NOT have been performed. 3. The Certification Authority Reference SHALL has been read from the EF.CVCA file or PACE protocol (Primary trust point) . 4. All APDUs are sent as valid Secure Messaging APDUs. 5. All response data SHALL be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>

	3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. The key $PK_{DH,IFD}$ generated during PACE with Chip Authentication Mapping in preconditions SHALL be used. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response
Post conditions	1. None

3.9.18 Test case ISO7816_K_16

Purpose	Positive test with a valid terminal authentication process after PACE-CAM and additional Chip Authentication
Version	3.00
References	[R1] Part 11 7.1
Profile	CA, CA-LDS1, TA, TA-LDS1, PACE, PACE-CAM
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed with PACE-CAM. Note PACE SHALL be used, the chip's ephemeral PACE public key SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. $ID_{IC} = Comp(PK_{DH,IC})$ 2. The Chip Authentication mechanism SHALL have been performed. 3. The Certification Authority Reference SHALL has been read from the EF.CVCA file or PACE protocol (Primary trust point). 4. All APDUs are sent as valid Secure Messaging APDUs. 5. All response data SHALL be SM protected.

Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. The newest key $PK_{DH,IFD}$ generated during Chip Authentication after PACE with Chip Authentication Mapping in preconditions SHALL be used. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response
Post conditions	1. None

3.9.19 Test case ISO7816_K_17

Purpose	Negative test with a valid terminal authentication process after PACE-CAM and additional Chip Authentication
Version	3.00
References	[R1] Part 11 7.1

Profile	CA, CA-LDS1, TA, TA-LDS1, PACE, PACE-CAM
Preconditions	1. The "Open LDS Application" procedure SHALL have been performed with PACE-CAM. Note PACE SHALL be used, the chip's ephemeral PACE public key SHALL be used to build the encrypted terminal signature (S_{PCD}) for the External Authenticate command. $ID_{IC} = Comp(PK_{DH,IC})$ 2. The Chip Authentication mechanism SHALL have been performed. 3. The Certification Authority Reference SHALL has been read from the EF.CVCA file or PACE protocol (Primary trust point). 4. All APDUs are sent as valid Secure Messaging APDUs. 5. All response data SHALL be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the T-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' The encrypted terminal signature SHALL be computed with ID_{IC} as defined in 3.9.2. The first key $PK_{DH,IFD}$ generated during PACE with Chip Authentication Mapping in preconditions SHALL be used. <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response

	3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. ISO checking error or ISO execution error or SW '63 00' in an SM response
Post conditions	1. None

3.10 Unit Test ISO7816_L – Effective Access Conditions

3.10.1 Introduction

This unit tests evaluation of the effective access conditions which has to be done by the chip. The chip has to grant access to sensitive data only if the complete terminal authentication mechanism has been performed. Furthermore, the access to the specific data groups depends on the access condition flags encoded in the DV and IS certificate.
All test cases of this test unit which require the "Open ePassport Application" procedure SHALL be performed twice (one test run with BAC and one with PACE) if the chip supports both protocols. If the chip only supports one of these protocols (BAC or PACE), only one test run has to be performed with the supported protocol used in the "Open ePassport Application" procedure.

3.10.2 Preconditions

The preconditions depend on the applicable profiles defined in clause 2.2.

3.10.2.1 Preconditions for TA-EU

Apply preconditions defined below:

1. The "Open LDS Application" procedure (see subclause 2.1) SHALL have been performed.
2. The Chip Authentication mechanism SHALL have been performed as well.
3. The Certification Authority Reference SHALL has been read from the EF.CVCA file (Primary trust point).
4. All APDUs are sent as valid Secure Messaging APDUs.
5. All response data SHALL be SM protected.

3.10.2.2 Preconditions for TA-LDS1

To be defined.

3.10.2.3 Preconditions for TA-MF

To be defined.

3.10.3 Test case ISO7816_L_1

Purpose	Positive test with a valid terminal authentication process with access permission for DG 3 if the DV certificate permits access to DG 3 and DG 4 while the IS certificate enables only the access to DG 3.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG3
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	<Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. '<data group 3 content data> 90 00' in an SM response
Post conditions	1. None

3.10.4 Test case ISO7816_L_2

Purpose	Negative test - Test that data group 4 cannot be accessed if the DV certificate permits access to DG 3 and DG 4 while the IS certificate enables only the access to DG 3.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG4
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file.

	2. Send the appropriate DV-Certificate as specified in DV_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 4 has NOT been granted. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.5 Test case ISO7816_L_3

Purpose	Positive test with a valid terminal authentication process with access permission for DG 4 if the DV certificate permits access to DG 3 and DG 4 while the IS certificate enables only the access to DG 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1

Profile	TA, DG4
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_2. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 4. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 4 has been granted. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response

	8. '<data group 4 content data> 90 00' in an SM response
Post conditions	1. None

3.10.6 Test case ISO7816_L_4

Purpose	Negative test - Test that data group 3 cannot be accessed if the DV certificate permits access to DG 3 and DG 4 while the IS certificate enables only the access to DG 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG3
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_2. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 4. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02.

	8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.7 Test case ISO7816_L_5

Purpose	Positive test with a valid terminal authentication process for DG 3 if the DV certificate grant access to data group 3 only and the IS certificate enable access to both data 3 and 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG3
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_2. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 only. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_3. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'

	7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. '<data group 3 content data> 90 00' in an SM response
Post conditions	1. None

3.10.8 Test case ISO7816_L_6

Purpose	Negative test - Test that data group 4 cannot be accessed if the DV certificate grant access to data group 3 only and the IS certificate enable access to both data 3 and 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG4
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_2. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 only. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3 and 4.

	5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 4 has NOT been granted. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.9 Test case ISO7816_L _7

Purpose	Positive test with a valid terminal authentication process for DG 4 if the DV certificate grant access to data group 4 only and the IS certificate enable access to both data 3 and 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG4
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 4 only. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	<Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 4 has been granted. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. '<data group 4 content data> 90 00' in an SM response
Post conditions	1. None

3.10.10 Test case ISO7816_L_8

Purpose	Negative test - Test that data group 3 cannot be accessed if the DV certificate grant access to data group 4 only and the IS certificate enable access to both data 3 and 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG3
Preconditions	1. See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file.

	2. Send the appropriate DV-Certificate as specified in DV_CERT_2_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 4 only. 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.11 Test case ISO7816_L_9

Purpose	Negative test - This test verifies that a successful certificate chain validation without external authenticate does not enable the access to the sensitive data in data group 3.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1

Profile	TA, DG3
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the "Certificate Set 1" chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.12 Test case ISO7816_L_10

Purpose	Negative test - This test verifies that a successful certificate chain validation

	without external authenticate does not enable the access to the sensitive data in data group 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG4
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> he appropriate IS-Certificate as specified in the “Certificate Set 1” chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 4. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 5. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 6. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 4 has NOT been granted. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.13 Test case ISO7816_L_11

Purpose	Negative test - Test with a failed external authenticate command does not enable the access to the sensitive data in data group 3.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG3
Preconditions	See 3.10.2

Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the “Certificate Set 1” chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. The last byte of the signature is changed to make it invalid 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response

	7. Execution error or ISO checking error or warning error '63 00' in an SM response 8. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.14 Test case ISO7816_L_12

Purpose	Negative test - Test with a failed external authenticate command does not enable the access to the sensitive data in data group 4.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG4
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the “Certificate Set 1” chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.

	The last byte of the signature is changed to make it invalid 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 4 has NOT been granted. '0C B0 84 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. Execution error or ISO checking error or warning error '63 00' in an SM response 8. Execution error or ISO checking error in an SM response
Post conditions	1. None

3.10.15 Test case ISO7816_L_13

Purpose	Negative test - Test that the chip rejects to fall back to BAC secure messaging after terminal has been authenticated as extended inspection system
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, BAC, MF
Preconditions	The LDS application SHALL have been selected. The BAC mechanism SHALL have been performed. The Chip Authentication mechanism SHALL have been performed as well. The Certification Authority Reference SHALL have been read from the EF.CVCA file (Primary trust point).
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the “Certificate Set 1” chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>

	5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. 8. Select the MF by sending the given Select APDU to the eMRTD: '0C A4 00 0C 0A 8E 08 <Checksum> 00' 9. Under SM, Perform an additional run of the "Open ePassport Application" procedure with BAC 10. Use new SM keys from BAC run. Send the READ BINARY (with SFI) command to the eMRTD to verify the access to Data Group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. '90 00' in an SM response, or ISO checking error, or ISO checking error in an SM response. If this step returns an ISO checking error the next test steps SHALL be skipped. 9. Successful run of "Open ePassport Application" procedure or ISO checking error. If "Open ePassport Application" procedure fails, the next steps SHALL be skipped. 10. ISO checking error in an SM response (chip SHALL reject access to Data Group 3 since access condition from previous Terminal Authentication SHALL be reset).
Post conditions	1. None

3.10.16 Test case ISO7816_L_14

Purpose	Negative test - Test that the chip rejects a second PACE run or reset extended access rights after successful second PACE run.
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, PACE
Preconditions	The "Open ePassport Application" procedure SHALL have been performed. PACE SHALL be used. The Chip Authentication mechanism SHALL have been performed as well. The Certification Authority Reference SHALL have been read from the EF.CVCA file (Primary trust point).

Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the "Certificate Set 1" chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. 8. Select the MF by sending the given Select APDU to the eMRTD: '0C A4 00 0C 0A 8E 08 <Checksum> 00' 9. Under SM, Perform a second run of the "Open ePassport Application" procedure with PACE. 10. Use new SM keys from second PACE run. Send the READ BINARY (with SFI) command to the eMRTD to verify the access to data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted terminal generated authentication token
Expected results	1. '90 00' in an SM response

	2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. Execution error or ISO checking error or warning error '63 00' in an SM response 8. '90 00' in an SM response, or Execution error or ISO checking error, or Execution error or ISO checking error in a SM response. If this step returns an Execution error or ISO checking error the next test steps SHALL be skipped. 9. Successful run of "Open ePassport Application" procedure or Execution error or ISO checking error. If "Open ePassport Application" procedure fails, the next steps SHALL be skipped. 10. Execution error or ISO checking error in a SM response (chip SHALL reject access to data group 3 since access condition from previous Terminal Authentication SHALL be reset)
Post conditions	1. None

3.10.17 Test case ISO7816_L_15

Purpose	Negative test - Test that the chip rejects an additional BAC run or resets extended access rights after the terminal has been authenticated as extended inspection system with a first PACE run
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, PACE, BAC
Preconditions	The "Open ePassport Application" procedure SHALL have been performed. PACE SHALL be used. The Chip Authentication mechanism SHALL have been performed as well. The Certification Authority Reference SHALL have been read from the EF.CVCA file (Primary trust point).
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used.

	4. Send the appropriate IS-Certificate as specified in the "Certificate Set 1" chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01. 8. Select the MF by sending the given Select APDU to the eMRTD: '0C A4 00 0C 0A 8E 08 <Checksum> 00' 9. Under SM, Perform an additional run of the "Open ePassport Application" procedure with BAC. 10. Use new SM keys from BAC run. Send the READ BINARY (with SFI) command to the eMRTD to verify the access to data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted terminal generated authentication token
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. Execution error or ISO checking error or warning error '63 00' in an SM response 8. '90 00' in an SM response, or Execution error or ISO checking error, or Execution error or ISO checking error in a SM response. If this step returns an Execution error or ISO checking error the next test steps SHALL be skipped. 9. Successful run of "Open ePassport Application" procedure or Execution error or ISO checking error. If "Open ePassport Application" procedure fails, the next steps SHALL be skipped. 10. Execution error or ISO checking error in a SM response (chip SHALL reject access to data group 3 since access condition from previous Terminal Authentication SHALL be reset)
Post conditions	1. None

3.10.18 Test case ISO7816_L_16

Purpose	Negative test - Test that the chip rejects an additional BAC run or resets extended access rights after the terminal has been authenticated as extended inspection system with a first BAC run
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, PACE, BAC
Preconditions	The "Open ePassport Application" procedure SHALL have been performed. BAC SHALL be used. The Chip Authentication mechanism SHALL have been performed as well. The Certification Authority Reference SHALL have been read from the EF.CVCA file (Primary trust point).
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the “Certificate Set 1” chapter as T_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.

	8. Select the MF by sending the given Select APDU to the eMRTD: '0C A4 00 0C 0A 8E 08 <Checksum> 00' 9. Under SM, Perform an additional run of the "Open ePassport Application" procedure with BAC. 10. Use new SM keys from PACE run. Send the READ BINARY (with SFI) command to the eMRTD to verify the access to data group 3 has NOT been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted terminal generated authentication token
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. Execution error or ISO checking error or warning error '63 00' in an SM response 8. '90 00' in an SM response, or Execution error or ISO checking error, or Execution error or ISO checking error in a SM response. If this step returns an Execution error or ISO checking error the next test steps SHALL be skipped. 9. Successful run of "Open ePassport Application" procedure or Execution error or ISO checking error. If "Open ePassport Application" procedure fails, the next steps SHALL be skipped. 10. Execution error or ISO checking error in a SM response (chip SHALL reject access to data group 3 since access condition from previous Terminal Authentication SHALL be reset)
Post conditions	1. None

3.10.19 Test case ISO7816_L_17

Purpose	Negative test - Test that the chip rejects an additional Terminal Authentication
Version	3.00
References	[R1] Part 11 7.1 & annex K.1
Profile	TA, DG3
Preconditions	See 3.10.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as read from the EF.CVCA file. 2. Send the appropriate DV-Certificate as specified in DV_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This DV-Certificate grants access to data group 3 and 4.

	3. Send the given MSE: Set DST APDU to the eMRTD. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in T_CERT_2_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This IS-Certificate grants only access to data group 3. 5. Send the given MSE: Set AT APDU to the eMRTD. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Holder Reference > The Certification Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eMRTD. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eMRTD. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_02. 8. Send the given READ BINARY (with SFI) command to the eMRTD, to verify the access to the data group 3 has been granted. '0C B0 83 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in an SM response 2. '90 00' in an SM response 3. '90 00' in an SM response 4. '90 00' in an SM response 5. '90 00' in an SM response 6. '<Eight bytes of random data> 90 00' in an SM response 7. '90 00' in an SM response 8. '<data group 3 content data> 90 00' in an SM response
Post conditions	1. None

3.11 Unit Test ISO7816_M – Update mechanism

3.11.1 Introduction

This unit contains all test cases which update the chips persistent memory. Therefore, these tests can be performed only once with a combination of a distinct sample and set of certificates. To reproduce this test unit, a new set with future certificate dates has to be created or a different test object has to be used. The certificates used in this Unit Test are defined in Certificate Sets 3 and 4 of this document. The references used in the test cases are generic and not linked to any application.

3.11.2 Preconditions

The preconditions depend on the applicable profiles defined in clause 2.2.

3.11.2.1 Preconditions for TA-EU

Apply preconditions defined below:

1. The "Open LDS Application" procedure (see subclause 2.1) SHALL have been performed.
2. The Chip Authentication mechanism SHALL have been performed as well.
3. The Certification Authority Reference SHALL has been read from the EF.CVCA file (Primary trust point).
4. All APDUs are sent as valid Secure Messaging APDUs.
5. All response data SHALL be SM protected

3.11.2.2 Preconditions for TA-LDS1

To be defined.

3.11.2.3 Preconditions for TA-MF

To be defined.

3.11.3 Test case ISO7816_M_1

Purpose	Test the “Current Date” update mechanism with a new domestic IS certificate.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.11.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.10.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_3_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The DV certificate is marked as a domestic certificate. 3. Send the MSE: Set DST APDU the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L83> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_3_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects

	7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate has an advanced effective date. Since the DV certificate was marked as a domestic one, the chip SHALL update the current date. Reset the chip after this step and restore the preconditions for this test case BEFORE the next step is performed. 5. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.10.2. 6. Send the appropriate DV-Certificate as specified in DV_CERT_3_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The DV certificate is marked as a domestic certificate. 7. Send the MSE: Set DST APDU the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the DV-Certificate sent in step 6 has to be used. 8. Send the appropriate T-Certificate as specified in T_CERT_3_2. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate has an expiry date BEFORE the effective of the T-certificate used in step 4. Therefore this certificate SHALL be rejected.
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' in a valid secure messaging response. 6. '90 00' in a valid secure messaging response. 7. '90 00' in a valid secure messaging response. 8. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. This certificate SHALL no longer be valid, since the current date of the chip has been updated.
Post conditions	None

3.11.4 Test case ISO7816_M_2

Purpose	Test the “Current Date” update mechanism with a new DV certificate.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.11.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference>

	The Certification Authority Reference SHALL be used as defined in 3.10.2. 2. Send the appropriate DV-Certificate as specified in DV_CERT_3_2. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> The DV certificate has an advanced effective data beyond the expiration date of DV_CERT_5_1. Reset the chip after this step and restore the preconditions for this test case before the next steo is performed. 3. Send the MSE: Set DST APDU the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.10.2. 4. Send the appropriate DV-Certificate as specified in DV_CERT_3_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> This certificate has an expiration date before the effective date that was set in step 2. Therefore, this certificate SHALL be rejected.
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. This certificate SHALL no longer be valid, since the current date of the chip has been updated.
Post conditions	None

3.11.5 Test case ISO7816_M_3

Purpose	Test the “Trust Point” update mechanism with a new link certificate.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.11.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.10.2. 2. Send the appropriate link Certificate as specified LINK_CERT_3_1. The eMRTD SHALL update the trust point with this new certificate. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Power down the field or remove the passport from the reader, so that the chip looses all temporary information. This is done to prove, that the new trust point has been stored in persistent memory. Power up the chip

Reestablish the preconditions
Read the EF.CVCA as exactly 36 bytes
'0C A4 02 0C 15 87 09 01 <Cryptogram> 8E 08 <Checksum> 00'
The cryptogram contains the encrypted fileID of the EF.CVCA file <fid.EF.CVCA>
'0C B0 00 00 0D 97 01 24 8E 08 <Checksum> 00'
Check that EF.CVCA file contains now two trust points and verify that the new trust point is at the first position and the previous one has been moved to the second position.
Any remaining bytes of the EF.CVCA content SHALL be filled with '00'

4. Send the MSE: Set DST APDU the eMRTD.
'0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'
<Cryptogram> contains the following encrypted data objects
83 <L_{83}> <certificate authority reference>
The Certification Authority Reference SHALL be the SECOND trust point as defined in 3.10.2.
5. Send the appropriate DV-Certificate as specified in DV_CERT_3_2.
'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>'
<Cryptogram> contains the following encrypted data objects
7F 4E <L_{7F4E}> <certificate body>
5F 37 <L_{5F37}> <certificate signature>
Since the previous trust point is still valid, the certificate SHALL be verified successfully.
Reset the chip after this step and restore the preconditions for this test case before the next step is performed.
6. Send the given MSE: Set DST APDU to the eMRTD
'0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'
<Cryptogram> contains the following encrypted data objects
83 <L_{83}> <certificate authority reference>
The Certification Authority Reference SHALL be the FIRST trust point as defined in 3.10.2.
7. Send the appropriate DV-Certificate as specified in DV_CERT_3_3.
'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>'
<Cryptogram> contains the following encrypted data objects
7F 4E <L_{7F4E}> <certificate body>
5F 37 <L_{5F37}> <certificate signature>
Since the effective date of this certificate is after the expiration date of the original trust point, the chip SHALL update the current date and SHALL also disable the original trust point.
Reset the chip after this step and restore the preconditions for this test case before the next step is performed
8. Send the MSE: Set DST APDU the eMRTD.
'0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'
<Cryptogram> contains the following encrypted data objects
83 <L_{83}> <certificate authority reference>
Use the original Certification Authority Referenc (same as in step4).
9. Send the appropriate DV-Certificate as specified in DV_CERT_3_2.
'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>'
<Cryptogram> contains the following encrypted data objects
7F 4E <L_{7F4E}> <certificate body>
5F 37 <L_{5F37}> <certificate signature>

	Since the trust point has been disabled, the certificate verification SHALL fail.
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. True 4. '90 00' in a valid secure messaging response. 5. '90 00' in a valid secure messaging response. 6. '90 00' in a valid secure messaging response. 7. '90 00' in a valid secure messaging response. 8. '90 00' or ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. 9. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response. This certificate SHALL no longer be valid, since the current date of the chip has been updated.
Post conditions	None

3.11.6 Test case ISO7816_M_4

Purpose	Test the "Trust Point" update mechanism with two link certificates.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.11.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be the FIRST trust point as defined in 3.10.2. 2. Send the appropriate link certificate as specified in LINK_CERT_3_2. The eMRTD SHALL update the trust point with this new certificate. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Reference SHALL be used as specified in the Link certificate used in step 2. 4. Send the appropriate link certificate as specified in LINK_CERT_4_1. The eMRTD SHALL update the trust point with this new certificate. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Reference SHALL be used as specified in the second Link certificate used in step 4.

	6. Send the appropriate DV-Certificate as specified in DV_CERT_4_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 7. Read the file EF.CVCA as exactly 36 bytes '0C A4 02 0C 15 87 09 01 <Cryptogram> 8E 08 <Checksum> 00' The cryptogram contains the encrypted fileID of the EF.CVCA file <fid.EF.CVCA> '0C B0 00 00 0D 97 01 24 8E 08 <Checksum> 00' Verify the EF.CVCA that both new trust points are present. The previous trust point from the LINK_CERT_3_1 SHALL be gone. The remaining (three) bytes of the EF.CVCA content SHALL be padded with '00'
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' in a valid secure messaging response. 6. '90 00' in a valid secure messaging response. 7. True
Post conditions	None

3.11.7 Test case ISO7816_M_5

Purpose	Test the transition CVCA ⇨ CVCA ⇨ IS
Version	3.00
References	[R1] Part 11 7.1
Profile	TA
Preconditions	See 3.11.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be used as defined in 3.10.2 (Primary trust point). 2. Send the appropriate CVCA-Certificate as specified in LINK_CERT_4_2. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the given MSE: Set DST APDU the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Holder Reference stored inside the new CVCA-Certificate sent in step 2 has to be used. 4. Send the appropriate T-Certificate as specified in T_CERT_4. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response.

	2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	1. None

3.11.8 Test case ISO7816_M_6

Purpose	Test sanity of the EF.CVCA
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, TA-EU
Preconditions	See 3.11.2
Test scenario	1. Send the given SELECT APDU to the eMRTD '0C A4 02 0C 15 87 09 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the encrypted fileID of the EF.CVCA file <fid.EF.CVCA> 2. Send the given READ BINARY APDU to the eMRTD trying to read 36 bytes. '0C B0 00 00 0D 97 01 24 8E 08 <Checksum> 00' 3. The EF.CVCA SHALL contain two trust points. 4. The remaining (six) bytes of the EF.CVCA content SHALL be padded with '00'. 5. Send another READ BINARY APDU to the eMRTD trying to read another byte at offset 36. '0C B0 00 24 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' in a valid secure messaging response. 2. Exactly 36 bytes of content data and '90 00' in a valid secure messaging response. 3. True. 4. True 5. ISO_checking_error or ISO_execution_error or ISO_warning in a valid secure messaging response.
Post conditions	1. None

3.12 Unit Test ISO7816_N – Migration policies

3.12.1 Introduction

This unit covers all tests about the migration policies. This mechanism is used for the import of new CVCA key with new TA algorithm in post issuance phase.
The purpose of this unit is to ensure the migration policy(ies) claimed by the manufacturer can be implemented.
This unit has to be performed once for each possible migration scenario indicated by the eMRTD provider. After the algorithm has been updated, the full test specification has to be repeated based on this new algorithm.
The certificates used in this Unit Test are defined in Certificate Set 5 of this document. The references used in the test cases are generic and not linked to any application.

3.12.2 Preconditions

The preconditions depend on the applicable profiles defined in clause 2.2.

3.12.2.1 Preconditions for TA-EU

Apply preconditions defined below:

1. The "Open LDS Application" procedure SHALL have been performed.
2. The Chip Authentication mechanism SHALL have been performed as well.
3. The Certification Authority Reference SHALL has been read from the EF.CVCA file (Primary trust point).
4. All APDUs are sent as valid Secure Messaging APDUs.
5. All response data SHALL be SM protected

3.12.2.2 Preconditions for TA-LDS1

To be defined.

3.12.2.3 Preconditions for TA-MF

To be defined.

3.12.3 Test case ISO7816_N_1

Purpose	Test mechanism migration according to the manufacturer's implementation statement.
Version	3.00
References	[R1] Part 11 7.1
Profile	TA, MIG
Preconditions	See 3.12.2
Test scenario	1. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Authority Reference SHALL be the FIRST trust point as defined in 3.12.2. 2. Send the appropriate link certificate with the updated mechanism as defined in LINK_CERT_5. The eMRTD SHALL update the trust point with this new certificate. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 3. Send the MSE: Set DST APDU the eMRTD. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects

	83 <L_{83}> <certificate authority reference> The Certification Reference SHALL be used as specified in the Link certificate used in step 2. The chip SHALL be able to use the updated cryptographic algorithms as introduced by the link certificate in step 2. 4. Send the appropriate DV-Certificate as specified DV_CERT_5. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 5. Send the given MSE: Set DST APDU to the eMRTD '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <certificate authority reference> The Certification Reference SHALL be used as specified in the DV-Certificate used in step 4. 6. Send the appropriate T-Certificate as specified in T_CERT_5. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' in a valid secure messaging response. 2. '90 00' in a valid secure messaging response. 3. '90 00' in a valid secure messaging response. 4. '90 00' in a valid secure messaging response. 5. '90 00' in a valid secure messaging response. 6. '90 00' in a valid secure messaging response.
Post conditions	None

3.13 Unit Test ISO7816_O - Security Conditions for PACE-protected eMRTDs

3.13.1 Introduction

This unit tests the security conditions of a PACE-protected eMRTD. It SHALL NOT be possible to select and read the content of any present file. The tests of this unit try to access the files with an explicit SelectFile command, a ReadBinary command with implicit file selection via the short file identifier (SFI), and unsecured ReadBinary while access is granted.
The tests in this unit only apply to PACE-protected eMRTDs (profile PACE).
The tests in this unit do not test the SM implementation including postconditions (e.g. SM termination); therefore, status bytes MAY be returned in SM or without it. Unit ISO7816_P handles this. In the following test cases, the term "PACE protocol is granted" means that the inspection system has successfully authenticated to the eMRTD. The first PACEInfo data structure in the EF.CardAccess has to be used.
Note: Unsecured SelectApplication command in this Test Unit can return '6982' or '9000'. According to [R1] Part 11, PACE protocol is implemented in addition to Basic Access Control. The unsecured SelectApplication command returns '9000' in this case. eMRTD supporting only PACE without Basic Access Control could return '6982' on unsecured SelectApplication command.
Note that if nothing is mentioned, unsecured context is applied.

Note: when accessing to protected DG by EAC, extended Access control SHALL be granted.

3.13.2 Test Case ISO7816_O_1

Purpose	Accessing the EF.COM file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	3. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1E'
Expected results	3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 4. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.3 Test Case ISO7816_O_2

Purpose	Accessing the EF.SOD file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1D'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.4 Test Case ISO7816_O_3

Purpose	Accessing the EF.DG1 file with explicit file selection

Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 01'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.5 Test Case ISO7816_O_4

Purpose	Accessing the EF.DG2 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 02'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.6 Test Case ISO7816_O_5

Purpose	Accessing the EF.DG3 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	(PACE, EAC, DG3) or (PACE, DG3)
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 03'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.7 Test Case ISO7816_O_6

Purpose	Accessing the EF.DG4 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	(PACE, EAC, DG4) or (PACE, DG4)
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 04'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.8 Test Case ISO7816_O_7

Purpose	Accessing the EF.DG5 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG5
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 05'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.9 Test Case ISO7816_O_8

Purpose	Accessing the EF.DG6 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG6
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 06'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.10 Test Case ISO7816_O_9

Purpose	Accessing the EF.DG7 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG7

Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 07'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.11 Test Case ISO7816_O_10

Purpose	Accessing the EF.DG8 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG8
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 08'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.12 Test Case ISO7816_O_11

Purpose	Accessing the EF.DG9 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG9
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 09'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.13 Test Case ISO7816_O_12

Purpose	Accessing the EF.DG10 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG10
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 0A'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.14 Test Case ISO7816_O_13

Purpose	Accessing the EF.DG11 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG11
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 0B'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.15 Test Case ISO7816_O_14

Purpose	Accessing the EF.DG12 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG12
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 0C'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.16 Test Case ISO7816_O_15

Purpose	Accessing the EF.DG13 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11

Profile	PACE, DG13
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 0D'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.17 Test Case ISO7816_O_16

Purpose	Accessing the EF.DG14 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 0E'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.18 Test Case ISO7816_O_17

Purpose	Accessing the EF.DG15 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, AA
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 0F'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.19 Test Case ISO7816_O_18

Purpose	Accessing the EF.DG16 file with explicit file selection
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG16
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01'

	2. Send the following SelectApplication APDU to the eMRTD '00 A4 02 0C 02 01 10'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.20 Test Case ISO7816_O_19

Purpose	Accessing the EF.COM file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 9E 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.21 Test Case ISO7816_O_20

Purpose	Accessing the EF.SOD file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 9D 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.22 Test Case ISO7816_O_21

Purpose	Accessing the EF.DG1 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01'

	2. Send the following ReadBinary APDU to the eMRTD '00 B0 81 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.23 Test Case ISO7816_O_22

Purpose	Accessing the EF.DG2 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 82 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.24 Test Case ISO7816_O_23

Purpose	Accessing the EF.DG3 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	(PACE,EAC, DG3) or (PACE, DG3)
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 83 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.25 Test Case ISO7816_O_24

Purpose	Accessing the EF.DG4 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	(PACE, EAC, DG4) or (PACE, DG4)
Preconditions	1. Reset the chip

Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 84 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.26 Test Case ISO7816_O_25

Purpose	Accessing the EF.DG5 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG5
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 85 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.27 Test Case ISO7816_O_26

Purpose	Accessing the EF.DG6 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG6
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 86 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.28 Test Case ISO7816_O_27

Purpose	Accessing the EF.DG7 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG7
Preconditions	1. Reset the chip

Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 87 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.29 Test Case ISO7816_O_28

Purpose	Accessing the EF.DG8 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG8
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 88 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.30 Test Case ISO7816_O_29

Purpose	Accessing the EF.DG9 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG9
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 89 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.31 Test Case ISO7816_O_30

Purpose	Accessing the EF.DG10 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG10
Preconditions	1. Reset the chip

Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 8A 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.32 Test Case ISO7816_O_31

Purpose	Accessing the EF.DG11 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG11
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 8B 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.33 Test Case ISO7816_O_32

Purpose	Accessing the EF.DG12 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG12
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 8C 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.34 Test Case ISO7816_O_33

Purpose	Accessing the EF.DG13 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG13
Preconditions	1. Reset the chip

Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 8D 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.35 Test Case ISO7816_O_34

Purpose	Accessing the EF.DG14 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 8E 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.36 Test Case ISO7816_O_35

Purpose	Accessing the EF.DG15 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, AA
Preconditions	1. Reset the chip
Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 8F 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.37 Test Case ISO7816_O_36

Purpose	Accessing the EF.DG16 file with implicit file selection (ReadBinary with SFI)
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG16
Preconditions	1. Reset the chip

Test scenario	1. Send the following SelectApplication APDU to the eMRTD '00 A4 04 0C 07 A0 00 00 02 47 10 01' 2. Send the following ReadBinary APDU to the eMRTD '00 B0 90 00 00'
Expected results	1. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. 2. Since the read access is prohibited without prior access control authentication, the response data field SHALL be empty. The eMRTD SHALL return status bytes '69 82' If SelectApplication in step 1 returned '69 82', this step is skipped

3.13.38 Test Case ISO7816_O_37

Purpose	Accessing the EF.COM file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.COM encoded as a valid SM to the eMRTD '0C B0 9E 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.39 Test Case ISO7816_O_38

Purpose	Accessing the EF.SOD file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.SOD encoded as a valid SM to the eMRTD '0C B0 9D 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.40 Test Case ISO7816_O_39

Purpose	Accessing the EF.DG1 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11

Profile	PACE
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG1 encoded as a valid SM to the eMRTD '0C B0 81 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.41 Test Case ISO7816_O_40

Purpose	Accessing the EF.DG2 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG2 encoded as a valid SM to the eMRTD '0C B0 82 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.42 Test Case ISO7816_O_41

Purpose	Accessing the EF.DG3 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	(PACE, EAC, DG3) or (PACE, DG3)
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected 3. The Extended Access Control SHALL be granted if necessary
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG3 encoded as a valid SM to the eMRTD '0C B0 83 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.43 Test Case ISO7816_O_42

Purpose	Accessing the EF.DG4 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.

Version	2.04
References	[R1] Part 10 & 11
Profile	(PACE, EAC, DG4) or (PACE, DG4)
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected 3. The Extended Access Control SHALL be granted if necessary
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG4 encoded as a valid SM to the eMRTD '0C B0 84 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.44 Test Case ISO7816_O_43

Purpose	Accessing the EF.DG5 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG5
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG5 encoded as a valid SM to the eMRTD '0C B0 85 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.45 Test Case ISO7816_O_44

Purpose	Accessing the EF.DG6 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG6
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG6 encoded as a valid SM to the eMRTD '0C B0 86 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.46 Test Case ISO7816_O_45

Purpose	Accessing the EF.DG7 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG7
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG7 encoded as a valid SM to the eMRTD '0C B0 87 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.47 Test Case ISO7816_O_46

Purpose	Accessing the EF.DG8 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG8
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG8 encoded as a valid SM to the eMRTD '0C B0 88 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.48 Test Case ISO7816_O_47

Purpose	Accessing the EF.DG9 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG9
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG9 encoded as a valid SM to the eMRTD '0C B0 89 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

	2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.49 Test Case ISO7816_O_48

Purpose	Accessing the EF.DG10 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG10
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG10 encoded as a valid SM to the eMRTD '0C B0 8A 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.50 Test Case ISO7816_O_49

Purpose	Accessing the EF.DG11 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG11
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG11 encoded as a valid SM to the eMRTD '0C B0 8B 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.51 Test Case ISO7816_O_50

Purpose	Accessing the EF.DG12 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG12
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG12 encoded as a valid SM to the eMRTD '0C B0 8C 00 0D 97 01 06 8E 08 <checksum> 00'

	2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.52 Test Case ISO7816_O_51

Purpose	Accessing the EF.DG13 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG13
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG13 encoded as a valid SM to the eMRTD '0C B0 8D 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.53 Test Case ISO7816_O_52

Purpose	Accessing the EF.DG14 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following "Read Binary (SFI)" APDU for EF.DG14 encoded as a valid SM to the eMRTD '0C B0 8E 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.54 Test Case ISO7816_O_53

Purpose	Accessing the EF.DG15 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, AA
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected

Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG15 encoded as a valid SM to the eMRTD '0C B0 8F 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.55 Test Case ISO7816_O_54

Purpose	Accessing the EF.DG16 file with ReadBinary. The test verifies the enforcement of SM after the PACE protocol has been performed successfully.
Version	2.04
References	[R1] Part 10 & 11
Profile	PACE, DG16
Preconditions	1. The PACE protocol SHALL have been performed using the MRZ-derived password. 2. The LDS application SHALL have been selected
Test scenario	1. Send the following “Read Binary (SFI)” APDU for EF.DG16 encoded as a valid SM to the eMRTD '0C B0 90 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.56 Test Case ISO7816_O_55

Purpose	Accessing the EF.CardSecurity file with explicit file selection
Version	2.08
References	[R1] Part 10 & 11
Profile	PACE, PACE-CAM
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1D'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.57 Test Case ISO7816_O_56

Purpose	Accessing the EF.CardSecurity file with implicit file selection (ReadBinary with SFI)
Version	2.08
References	[R1] Part 10 & 11
Profile	PACE, PACE-CAM
Preconditions	1. Reset the chip
Test scenario	1. Send the following ReadBinary APDU to the eMRTD '00 B0 9D 00 00'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.58 Test Case ISO7816_O_57

Purpose	Accessing the EF.CardSecurity file with ReadBinary. The test verifies the enforcement of SM after the PACE-CAM protocol has been performed successfully.

Version	2.08
References	[R1] Part 10 & 11
Profile	PACE, PACE-CAM
Preconditions	1. Reset the chip 2. The PACE protocol SHALL have been performed using the MRZ-derived password and PACE-CAM OID.
Test scenario	1. Send the following ReadBinary APDU to the eMRTD '0C B0 9D 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.13.59 Test Case ISO7816_O_58

Purpose	Accessing the EF.CardSecurity file with ReadBinary. The test verifies the enforcement of SM after a PACE protocol different from PACE-CAM has been performed successfully.
Version	2.08
References	[R1] Part 10 & 11
Profile	PACE, PACE-CAM
Preconditions	1. Reset the chip 2. The PACE protocol SHALL have been performed using the MRZ-derived password. PACE-CAM protocol SHALL NOT be selected
Test scenario	1. Send the following ReadBinary APDU to the eMRTD '0C B0 9D 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Read Binary APDU as an unsecured APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 2. The eMRTD SHALL return an ISO checking error or ISO execution error

3.14 Unit Test ISO7816_P – Password Authenticated Connection Establishment (PACE)

3.14.1 Introduction

This unit checks the PACE implementation of the eMRTD. The complete PACE access mechanism is tested, including robustness tests with invalid input data.
Since the tests in this unit apply to PACE-protected eMRTDs, they are only mandatory for eMRTDs complying with the PACE profile.
In case of PACE failure, [R1] Part 11 does not define clearly the conditions of use of BAC mechanism. This context is out of the scope of this test unit.

PACE establishes SM between an eMRTD and an inspection system based on weak (short) passwords. It enables the eMRTD to verify that the inspection system is authorized to access stored data and has the following features:

- Strong session keys are provided independent of the strength of the password.
- The entropy of the password(s) used to authenticate the inspection system can be very low (e.g. 6 digits are sufficient in general).

PACE uses keys K_π derived from passwords. For globally interoperable machine readable travel documents the following two passwords and corresponding keys are available as follows:

- **MRZ:** The key $K_\pi = KDF_\pi(MRZ)$ is REQUIRED. It is derived from the Machine Readable Zone (MRZ) similar to Basic Access Control, i.e. the key is derived from the Document Number, the Date of Birth and the Date of Expiry.
- **CAN:** The key $K_\pi = KDF_\pi(CAN)$ is OPTIONAL. It is derived from the Card Access Number (CAN). The CAN is a number printed on the *front side* of the data page

This unit SHALL be executed for each PACE protocol indicated in the PACEInfo elements present in the EF.CardAccess of the eMRTD. Pre-conditions SHALL be run with each PACEInfo elements.

Note: Unsecured SelectApplication command in this Test Unit can return '6982' or '9000'. According to [R1] Part 11, PACE protocol is implemented in addition to Basic Access Control. The unsecured SelectApplication command returns '9000' in this case. eMRTD supporting only PACE without Basic Access Control could return '6982' on unsecured SelectApplication command
Note that unsecured context is applied if nothing is mentioned.

3.14.2 Test Case ISO7816_P_1

Purpose	Positive test with a valid PACE protocol with MRZ password
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the Chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' If the current protocol is not PACE-CAM, skip the steps 6 and 7. 6. Read EF.CardSecurity with new derived SM keys. 7. Decrypt Encrypted Chip Authentication Data to recover CA data. Perform Key Agreement and verify keys 8. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD must return content of EF.CardSecurity. 7. Chip Authentication Data is correct according to [R1] Part 11 4.4.3.5.2 8. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.3 Test Case ISO7816_P_2

Purpose	Positive test with a valid PACE protocol with CAN password
Version	2.08

References	[R1] Part 11 4.4
Profile	PACE, PACE-CAN
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with CAN password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' If the current protocol is not PACE-CAM, skip the steps 6 and 7. 6. Read EF.CardSecurity with new derived SM keys. 7. Decrypt Encrypted Chip Authentication Data to recover CA data. Perform Key Agreement and verify keys 8. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD must return content of EF.CardSecurity. 7. Chip Authentication Data is correct according to [R1] Part 11 4.4.3.5.2 8. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.4 Test Case ISO7816_P_3

Purpose	Valid PACE protocol with MRZ password, but afterwards command without SM is used
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. Select LDS application under SM '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used 7. To verify the chip's ability to still require Secured APDU after performing valid PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response. 7. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.5 Void

Removed in version 2.11.

3.14.6 Test Case ISO7816_P_5

Purpose	MSE: Set AT command with an invalid data object tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 81 <L_{81}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.7 Test Case ISO7816_P_6

Purpose	MSE: Set AT command with an invalid PACE OID
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01' - <PACE OID> : {id-PACE 5} 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'

Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.8 Test Case ISO7816_P_7

Purpose	MSE: Set AT command with a PACE OID with tag '0x06'
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> {06 <$L_{PACE\ OID}$> <PACE OID>} 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.9 Test Case ISO7816_P_8

Purpose	MSE: Set AT command with a bad reference password
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 <Invalid password identifier> 84 <L_{84}> <private key reference>'

	- <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00'. The actual security operation which must fail using the wrong password reference in the MSE:Set AT command may be performed in a subsequent command. 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.10 Test Case ISO7816_P_9

Purpose	MSE: Set AT command with a private key reference unknown from the chip
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - Use a private key reference unknown from the chip 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. Select LDS application 7. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'

Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning or '90 00'. Steps 3 to 5 are skipped in case of ISO checking error or ISO execution error or ISO Warning. 3. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning or '90 00'. Steps 4 and 5 are skipped in case of ISO checking error or ISO execution error or ISO Warning. 4. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning or '90 00'. Step 5 is skipped in case of ISO checking error or ISO execution error or ISO Warning. 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning. 6. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error 7. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.11 Void

Removed in version v2.02

3.14.12 Test Case ISO7816_P_11

Purpose	General Authenticate to get the encrypted nonce command with a bad dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 8C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error.

	4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.13 Test Case ISO7816_P_12

Purpose	General Authenticate to get the encrypted nonce command without dynamic authentication data
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error. 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.14 Test Case ISO7816_P_13

Purpose	General Authenticate to get the encrypted nonce command with an additional object data
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 01 01 <Le>'

	3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.15 Test Case ISO7816_P_14

Purpose	Check PACE protocol with MRZ password after performing twice General Authenticate APDU to get the encrypted nonce.
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 4. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' Note: Mapping data SHALL be computed according to second generated nonce. 5. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 6. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 7. To verify the chip's ability to start the SM with the session keys, an arbitrary SM APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00'

	2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' with encrypted nonce different from step 2 or ISO checking error or ISO execution error. Remaining steps are skipped in case of error.2 4. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 5. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 6. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 7. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.16 Test Case ISO7816_P_15

Purpose	General Authenticate APDU to map the nonce with a bad dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 8C <L_{8C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO checking error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO

[2] Behavior depends on implementation

	execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.17 Test Case ISO7816_P_16

Purpose	General Authenticate APDU to map the nonce without a dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 81 <L_{81}> <Mapping Data> <Le>' 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO checking error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO checking error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.18 Test Case ISO7816_P_17

Purpose	General Authenticate APDU to map the nonce with a bad data object tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol

	- The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 82 <L_{82}> <Mapping Data> <Le>' 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.19 Void

Removed in version 2.11

3.14.20 Test Case ISO7816_P_19

Purpose	General Authenticate APDU to perform key agreement with a bad dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 8C <L_{8C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Select LDS application

	6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.21 Test Case ISO7816_P_20

Purpose	General Authenticate APDU to perform key agreement without dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.22 Test Case ISO7816_P_21

Purpose	General Authenticate APDU to perform key agreement with a bad data object tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly

Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> <Le>' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.23 Test Case ISO7816_P_22

Purpose	General Authenticate APDU to perform key agreement with an additional data object tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess.

	2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> {83 <L_{83}> <Ephemeral Public Key> 84 01 01} <Le>' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.24 Test Case ISO7816_P_23

Purpose	General Authenticate APDU to perform key agreement with invalid ephemeral public key (different key size)
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>'

	- The ephemeral public key SHALL be generated with domain parameters specifying a different key size (e.g. for a 224 bit key after mapping the nonce, a 192 bit ephemeral key pair is created) 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.25 Test Case ISO7816_P_24

Purpose	General Authenticate APDU to perform key agreement providing a (0,0) public key
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - The public key has to be coded as '04||x||y' where both x and y have a size according to the prime, but filled with '00' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is

	sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.26 Test Case ISO7816_P_25

Purpose	General Authenticate APDU to perform key agreement - test borderline cases for x- and y- coordinates (small x coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE , PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with an x-coordinate requiring at least one byte less than the length of P. Pad with leading zero bytes. Generate key pairs at random until a public key satisfying the constraint is obtained 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	- <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.27 Test Case ISO7816_P_26

Purpose	General Authenticate APDU to perform key agreement - test borderline cases for x- and y- coordinates (large x coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE, PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with an x-coordinate having its highest bit set to 1 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used

Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.28 Test Case ISO7816_P_27

Purpose	General Authenticate APDU to perform key agreement - test borderline cases for x- and y- coordinates (small y coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE , PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with a y-coordinate requiring at least one byte less than the length of P. Pad with leading zero bytes. Generate key pairs at random until a public key satisfying the constraint is obtained 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used

Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response

3.14.29 Test Case ISO7816_P_28

Purpose	General Authenticate APDU to perform key agreement - test borderline cases for x- and y- coordinates (large y coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE , PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with an y-coordinate having its highest bit set to 1 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00'

	2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.30 Test Case ISO7816_P_29

Purpose	General Authenticate APDU to perform key agreement – value higher than the prime
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-DH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with a wrong value (value larger than the Prime) ephemeral public key = prime p + 1 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note:

	In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.31 Test Case ISO7816_P_30

Purpose	General Authenticate APDU to perform key agreement – wrong point (value does not belong to the curve)
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with a wrong point (value does not belong to the curve) 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO

	execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.32 Test Case ISO7816_P_31

Purpose	General Authenticate APDU to perform Mutual Authenticate with a bad dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L80> <PACE OID> 83 01 01 84 <L84> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 8C <L_{7C}> 85 <L85> <Authentication Token> <Le>' 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.33 Test Case ISO7816_P_32

Purpose	General Authenticate APDU to perform Mutual Authenticate without dynamic authentication data tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.34 Test Case ISO7816_P_33

Purpose	General Authenticate APDU to perform Mutual Authenticate with a bad data object tag
Version	2.04
References	[R1] Part 11 4.4

Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 86 <L_{86}> <Authentication Token> <Le>' 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.35 Test Case ISO7816_P_34

Purpose	General Authenticate APDU to perform Mutual Authenticate with an additional data object tag
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly

Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> {85 <L_{85}> <Authentication Token> 86 01 01} <Le>' 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.36 Test Case ISO7816_P_35

Purpose	General Authenticate APDU to perform Mutual Authenticate with a wrong authentication token
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol

	- The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' - Replace the last byte of the Authentication token by its complementary. ex: replace 0x00 by 0xFF 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.37 Test Case ISO7816_P_36

Purpose	General Authenticate APDU to perform Mutual Authenticate with a shorter authentication token
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one

	parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' - Remove the last byte of the Authentication token. <L_{85}> must be correctly computed 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.38 Void

Removed in version 2.11.

3.14.39 Void

Removed in version 2.11.

3.14.40 Void

Removed in version 2.11.

3.14.41 Void

Removed in version 2.11.

3.14.42 Test Case ISO7816_P_41

Purpose	General Authenticate APDU to map the nonce with invalid ephemeral public key (different key size)
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)

Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - The ephemeral public key SHALL be generated with domain parameters specifying a different key size (e.g. if a 224 bit key is required, a 192 bit ephemeral key pair is created) 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.43 Test Case ISO7816_P_42

Purpose	General Authenticate APDU to map the nonce providing a (0,0) public key
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>'

	3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - The public key has to be coded as '04||x||y' where both x and y have a size according to the prime, but filled with '00' 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.44 Test Case ISO7816_P_43

Purpose	General Authenticate APDU to map the nonce - test borderline cases for x- and y- coordinates (small x coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE , PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Use an ephemeral public key with an x-coordinate requiring at least one byte less than the fewest bytes that can represent [$\log_{256} q$]. Pad with zero bytes. (For details on q see [R5]) 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM.

	'0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> 90 00' 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.45 Test Case ISO7816_P_44

Purpose	General Authenticate APDU to map the nonce - test borderline cases for x- and y- coordinates (large x coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Use an ephemeral public key with an x-coordinate having its highest bit set to 1 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used

Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.46 Test Case ISO7816_P_45

Purpose	General Authenticate APDU to map the nonce - test borderline cases for x- and y- coordinates (small y coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Use an ephemeral public key with a y-coordinate requiring at least one byte less than the fewest bytes that can represent $[\log_{256} q]$. Pad with zero bytes. (For details on q see [R5]) 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00'

	3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.47 Test Case ISO7816_P_46

Purpose	General Authenticate APDU to map the nonce - test borderline cases for x- and y- coordinates (large y coordinate)
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Use an ephemeral public key with an y-coordinate having its highest bit set to 1 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00'

	5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}>* *<Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.48 Test Case ISO7816_P_47

Purpose	General Authenticate APDU to map the nonce – value higher than the prime
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-DH, PACE-GM
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Use an ephemeral public key with a wrong value (value larger than the Prime) ephemeral public key = prime p + 1 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.49 Test Case ISO7816_P_48

Purpose	General Authenticate APDU to map the nonce – ephemeral mapping public key value is set to '00..00'
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-DH, PACE-GM
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly

Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Use an ephemeral public key with a value set to zero ('00..00' over the prime length) 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.50 Test Case ISO7816_P_49

Purpose	General Authenticate APDU to map the nonce – wrong point (value does not belong to the curve)
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>'

	- Use an ephemeral public key with a wrong point (value does not belong to the curve). An arbitrary generator can be used to generate the key pair.. 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return status bytes '90 00' or an ISO_Checking_Error or ISO execution error. Remaining steps are skipped in case of error. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.51 Test Case ISO7816_P_50

Purpose	General Authenticate APDU to map the nonce – wrong point encoding (the first byte of the public key is different from '01', '02', '03' and '04')
Version	2.08
References	[R1] Part 11 4.4
Profile	(PACE, PACE-ECDH, PACE-GM) or (PACE, PACE-CAM)
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - Send the ephemeral public encoded as follows : '77||x||y' 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00'

	2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.52 Test Case ISO7816_P_51

Purpose	General Authenticate APDU to map the nonce with invalid length for the additional nonce (size different from the expected one)
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-IM
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' - The nonce sent has a wrong length (different from the expected length) : length = length + 1 (byte 0x00 is added to the start of the byte string representing the nonce). 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.53 Test Case ISO7816_P_52

Purpose	General Authenticate APDU to perform key agreement - wrong point encoding (the first byte of the public key is different from '01', '02', '03' and '04')

Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-ECDH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Send the ephemeral public encoded as follows : '77||x||y' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.54 Test Case ISO7816_P_53

Purpose	General Authenticate APDU to perform key agreement – ephemeral public key value is set to '00..00'
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE, PACE-DH
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly

Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' - Use an ephemeral public key with a value set to zero ('00..00' over the prime length) 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.55 Test Case ISO7816_P_54

Purpose	General Authenticate to get the encrypted nonce command while the CLASS byte does not indicate command chaining
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one

	parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '00 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return '90 00' 2. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE MAY return status bytes '69 82'. Next step is skipped in case of returning '69 82'. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.56 Test Case ISO7816_P_55

Purpose	General Authenticate APDU to map the nonce while the CLASS byte does not indicate command chaining
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '00 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Select LDS application 5. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 4. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error.

	5. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.57 Test Case ISO7816_P_56

Purpose	General Authenticate APDU to perform key agreement while the CLASS byte does not indicate command chaining
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '00 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.58 Test Case ISO7816_P_57

Purpose	General Authenticate APDU to perform Mutual Authenticate while the CLASS byte indicates command chaining
Version	2.04
References	[R1] Part 11 4.4

Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '10 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return status bytes '68 83' 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.59 Test Case ISO7816_P_58

Purpose	General Authenticate APDU to perform Mutual Authenticate is not sent and replaced by another command
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>'

	- <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Replace the last command of the PACE protocol by the READ BINARY command: '00 B0 9C 00 01' 6. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return '7C <L_{7C}> 84 <L_{84}> < Ephemeral Public Key > 90 00' 5. The eMRTD SHALL return status bytes '90 00' or an ISO checking error or ISO execution error 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.60 Test Case ISO7816_P_59

Purpose	General Authenticate APDU to map the nonce while an unexpected command was executed between the nonce generation and the mapping
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess.

	2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the READ BINARY command: '00 B0 9C 00 01' 4. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 5. Select LDS application 6. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return status bytes '90 00' or an ISO_Checking_Error or ISO execution error 4. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 5. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.61 Test Case ISO7816_P_60

Purpose	General Authenticate APDU to perform the key agreement while an unexpected command was executed between the mapping and the key agreement
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the READ BINARY command: '00 B0 9C 00 01' 5. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>'

	6. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' - <Authentication Token> can be any valid dummy data. 7. Select LDS application 8. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return '90 00' or an ISO_Checking_Error or ISO execution error 5. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 6. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 7. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 8. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.62 Test Case ISO7816_P_61

Purpose	General Authenticate APDU to perform the mutual authentication while an unexpected command was executed between the key agreement and the mutual authentication
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>'

	5. Send the READ BINARY command: '00 B0 9C 00 01' 6. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 7. Select LDS application 8. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return '90 00' or an ISO checking error or ISO execution error 6. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 7. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 8. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.63 Test Case ISO7816_P_62

Purpose	General Authenticate APDU to perform Mutual Authenticate without using Tag 7F49 in the input for the authentication token
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>'

	4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' Use Tag 30 instead of Tag 7F49 6. Select LDS application 7. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 7. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.64 Void

Removed in version v2.0

3.14.65 Test Case ISO7816_P_64

Purpose	MSE: Set AT command without data object 80
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 83 01 01' 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>'

	6. Select LDS application 7. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning or '90 00'. Steps 3 to 5 are skipped in case of ISO checking error or ISO execution error or ISO Warning. 3. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning or '90 00'. Steps 4 and 5 are skipped in case of ISO checking error or ISO execution error or ISO Warning. 4. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning or '90 00'. Step 5 is skipped in case of ISO checking error or ISO execution error or ISO Warning. 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning. 6. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 7. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.66 Test Case ISO7816_P_65

Purpose	MSE: Set AT command with an empty data object 80
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 00 83 01 01' 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.67 Test Case ISO7816_P_66

Purpose	MSE: Set AT command with a too long data object 80.
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01' - <PACE OID> : invalid PACE OID which is too long (e.g.: '04 00 7F 00 07 02 02 04 02 02 02'). <L_{80}> is correctly computed. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.68 Test Case ISO7816_P_67

Purpose	MSE: Set AT command with a too short data object 80.
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01' - <PACE OID> : invalid PACE OID which is too short (e.g.: '04 00 7F 00 07 02 02 04 02'). <L_{80}> is correctly computed. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning

	3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.69 Test Case ISO7816_P_68

Purpose	MSE: Set AT command with a given PACE OID not supported by the eMRTD
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01' - <PACE OID> : valid PACE OID which is not supported by the eMRTD 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.70 Test Case ISO7816_P_69

Purpose	MSE: Set AT command without data object 83
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>'

	3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.71 Test Case ISO7816_P_70

Purpose	MSE: Set AT command with an empty data object 83
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 00 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.72 Test Case ISO7816_P_71

Purpose	MSE: Set AT command with an incorrect length byte for password reference (DO 83)
Version	2.04

References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 03 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.73 Test Case ISO7816_P_72

Purpose	MSE: Set AT command with a too long password reference (DO 83)
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 02 01 FF 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'

Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.74 Test Case ISO7816_P_73

Purpose	Positive test using domain parameter reference (DO 84) but eMRTD supports only one set of domain parameters
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. The PACE parameters were successfully read from EF.CardAccess. 3. EF.CardAccess contains one or more PACEInfo entries having all the same parameter Id.
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU even if the domain parameters are not ambiguous, 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.75 Test Case ISO7816_P_74

Purpose	Positive test without domain parameter reference (DO 84) and eMRTD supports only one set of domain parameters
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip

	2. The PACE parameters were successfully read from EF.CardAccess. 3. EF.CardAccess contains only one PACEInfo or more than one PACEInfo which are not ambiguous (only one parameter Id or distinct PACE OID).
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL NOT be included in the APDU 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' 6. To verify the chip's ability to start the SM with the session keys, the Select LDS command is sent to the chip under SM. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> *8A <L_{8A}> <Encrypted Chip Authentication Data>* 90 00' Data Object 8A shall be present only if the protocol is PACE-CAM. 6. The eMRTD SHALL return status bytes '90 00' in a valid SM response.

3.14.76 Test Case ISO7816_P_75

Purpose	MSE: Set AT command with an empty domain parameter reference (DO 84)
Version	2.05
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	This test can be performed in case 2 or more parameterId values have been assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 00' - <PACE OID> : Valid Object Identifier for the PACE protocol that has been assigned to 2 or more ParameterId values in EF.CardAccess 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>'

	3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO checking error or ISO execution error or '90 00' 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error. 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.77 Test Case ISO7816_P_76

Purpose	General Authenticate APDU to perform Mutual Authenticate with a longer authentication token
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>' - Extend the value of the Authentication token by one byte. <L_{85}> must be correctly computed 7. To verify the chip's ability to reject Secured APDU after performing incomplete PACE protocol, an arbitrary secured APDU is sent to the chip. '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' - <Cryptogram> contains eMRTD Application identifier 'A0 00 00 02 47 10 01' encrypted according to the SM being used
Expected results	1. The eMRTD SHALL return status bytes '90 00'

	2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' Note: In case of Integrated Mapping, <L_{82}> SHALL be set to '00' and < Mapping Data> SHALL be empty. 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 6. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response

3.14.78 Test Case ISO7816_P_77

Purpose	MSE: Set AT command with a PACE OID with tag '0x06'instead of Tag '0x80'
Version	2.04
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 06 <$L_{PACE\ OID}$> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Select LDS application 4. To verify the chip's ability to still require Secured APDU after performing incomplete PACE protocol, an arbitrary unsecured SM-requiring APDU is sent to the chip. '<Unsecured command as defined in table 1>'
Expected results	1. The eMRTD SHALL return an ISO_Checking_Error or ISO execution error 2. The eMRTD SHALL return an ISO checking error or ISO execution error or ISO_Warning 3. eMRTD supporting BAC and PACE SHALL return status bytes '90 00'. eMRTD supporting only PACE SHALL return an ISO checking error or ISO execution error or status bytes '90 00' instead. Next step is skipped in case of returning ISO_Checking_Error or ISO execution error . 4. The eMRTD SHALL return an ISO checking error or ISO execution error in an unsecured response.

3.14.79 Test Case ISO7816_P_78

Purpose	Positive test with a complete sequence of PACE without Chip Authentication Mapping commands with MRZ password. The tag 0x8A during PACE-GM and PACE-IM SHALL NOT be returned.
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE, PACE-CAM

Preconditions	1. Reset the Chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 01 84 <L_{84}> <private key reference>' - <PACE OID> : valid Object Identifier for PACE-GM or PACE-IM to the PACE protocol - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess. 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <Mapping Data> <Le>' 4. Send the given General Authenticate APDU to perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <Ephemeral Public Key> <Le>' 5. Send the given General Authenticate APDU to perform mutual authenticate: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <Authentication Token> <Le>'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return: '7C <L_{7C}> 80 <L_{80}> <encrypted nonce> 90 00' 3. The eMRTD SHALL return: '7C <L_{7C}> 82 <L_{82}> <Mapping Data> 90 00' 4. The eMRTD SHALL return: '7C <L_{7C}> 84 <L_{84}> <Ephemeral Public Key> 90 00' 5. The eMRTD SHALL return: '7C <L_{7C}> 86 <L_{86}> <Authentication Token> 90 00' - Tag 0x8A for Encrypted Chip Authentication Data SHALL NOT be present.

3.14.80 Test Case ISO7816_P_79

Purpose	Negative test to verify the Secure Messaging handling while PACE access is granted for the Select LDS application command (bad send sequence counter)
Version	2.11
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the Chip 2. EF.CardAccess has been read correctly 3. The PACE mechanism SHALL have been performed
Test scenario	1. Select LDS application During the coding of the SM APDU the SendSequenceCounter is not incremented. 2. To verify that the error in step 1 has terminated the SM session, send a valid SM APDU (Select LDS application) to the eMRTD.
Expected results	1. The eMRTD SHALL return status bytes '69 88' or '69 82'. 2. Since the session keys are no longer valid, the eMRTD SHALL return an ISO checking error or ISO execution error.

3.14.81 Test Case ISO7816_P_80

Purpose	Test to verify that the ephemeral public keys generated by the eMRTD are encoded as unsigned integer.
Version	3.00
References	[R1] Part 11 9.4.1
Profile	PACE, PACE-DH
Preconditions	1. Reset the Chip 2. EF.CardAccess has been read correctly 3. The PACE mechanism SHALL have been performed
Test scenario	1. Perform 300 times following steps: - Reset the Chip - Perform the PACE mechanism During the PACE mechanisms, the ephemeral public keys SHALL be encoded under unsigned integer. The minimum number of octets SHALL be used, i.e. leading octets of value 0x00 SHALL NOT be used.
Expected results	1. The eMRTD SHALL generate unsigned integer without leading octets of value 0x00 SHALL NOT be used during the 300 PACE mechanisms.

3.14.82 Test case ISO7816_P_81

Purpose	Negative test with a valid Password Authenticated Connection Establishment process using MRZ password, but afterwards a READ binary APDU command with corrupted checksum is used
Version	3.00
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <private key reference>' - <PACE OID> is a valid PACE OID fitting the implemented algorithm - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <mapping data> <Le>' 4. Perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 7C <L7C> 85 <L85> <authentication token> <Le>' 6. To verify that the correct SM is required, send the given READ Binary APDU for EF.CardAccess with corrupted checksum. The last byte of the checksum is incremented by one. '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00 with corrupted checksum' 7. To verify that the error in previous step has terminated the SM session, send the given READ Binary APDU for EF.CardAccess '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' 2. 7C <L_{7C}> '80' <L_{80}> <encrypted nonce> '90 00'

	3. 7C <L_{7C}> '82' <L_{82}> <mapping data> '90 00' 4. 7C <L_{7C}> '84' <L_{84}> <ephemeral public key> '90 00' 5. 7C <L_{7C}> '86' <L_{86}> <authentication token> '90 00' 6. ISO_checking_error or ISO_execution_error without SM 7. ISO_checking_error or ISO_execution_error without SM
Post conditions	1. None

3.14.83 Test case ISO7816_P_82

Purpose	Negative test with a valid Password Authenticated Connection Establishment process using MRZ password, but afterwards a SELECT APDU command with corrupted cryptogram is used
Version	3.00
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <private key reference >' - <PACE OID> is a valid PACE OID fitting the implemented algorithm - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <mapping data> <Le>' 4. Perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <authentication token> <Le>' 6. To verify that the correct SM is required, send the given SELECT APDU for EF.CardAccess with corrupted cryptogram. The last byte of the cryptogram is incremented by one. '0C A4 02 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00 with corrupted cryptogram' 7. To verify that the error in previous step has terminated the SM session, send the given READ Binary APDU for EF.CardAccess '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' 2. 7C <L_{7C}> '80' <L_{80}> <encrypted nonce> '90 00' 3. 7C <L_{7C}> '82' <L_{82}> <mapping data> '90 00' 4. 7C <L_{7C}> '84' <L_{84}> <ephemeral public key> '90 00' 5. 7C <L_{7C}> '86' <L_{86}> <authentication token> '90 00' 6. ISO_checking_error or ISO_execution_error without SM 7. ISO_checking_error or ISO_execution_error without SM
Post conditions	1. None

3.14.84 Test case ISO7816_P_83

Purpose	Negative test with a valid Password Authenticated Connection Establishment process using MRZ password, but afterwards a READ binary APDU command with bad Send Sequence Counter is used
Version	3.00

References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <private key reference>' - <PACE OID> is a valid PACE OID fitting the implemented algorithm - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <mapping data> <Le>' 4. Perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <authentication token> <Le>' 6. To verify that the correct SM is required, send the given READ Binary APDU for EF.CardAccess with bad Send Sequence Counter. During the coding of the SM APDU the Send Sequence Counter is not incremented. '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00 with bad SSC' 7. To verify that the error in previous step has terminated the SM session, send the given READ Binary APDU for EF.CardAccess '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' 2. 7C <L_{7C}> '80' <L_{80}> <encrypted nonce> '90 00' 3. 7C <L_{7C}> '82' <L_{82}> <mapping data> '90 00' 4. 7C <L_{7C}> '84' <L_{84}> <ephemeral public key> '90 00' 5. 7C <L_{7C}> '86' <L_{86}> <authentication token> '90 00' 6. ISO_checking_error or ISO_execution_error without SM 7. ISO_checking_error or ISO_execution_error without SM
Post conditions	1. None

3.14.85 Test case ISO7816_P_84

Purpose	Negative test with a valid Password Authenticated Connection Establishment process using MRZ password, but afterwards a READ binary APDU command with missing checksum is used
Version	3.00
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <private key reference>' - <PACE OID> is a valid PACE OID fitting the implemented algorithm - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess

	2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <mapping data> <Le>' 4. Perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <authentication token> <Le>' 6. To verify that the correct SM is required, send the given READ Binary APDU for EF.CardAccess with missing checksum '0C B0 (80 OR <sfi.EF.CardAccess>) 00 03 97 01 01 00 with missing checksum (Tag 8E is removed)' 7. To verify that the error in previous step has terminated the SM session, send the given READ Binary APDU for EF.CardAccess '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' 2. 7C <L_{7C}> '80' <L_{80}> <encrypted nonce> '90 00' 3. 7C <L_{7C}> '82' <L_{82}> <mapping data> '90 00' 4. 7C <L_{7C}> '84' <L_{84}> <ephemeral public key> '90 00' 5. 7C <L_{7C}> '86' <L_{86}> <authentication token> '90 00' 6. ISO_checking_error or ISO_execution_error without SM 7. ISO_checking_error or ISO_execution_error without SM
Post conditions	1. None

3.14.86 Test case ISO7816_P_85

Purpose	Negative test with a valid Password Authenticated Connection Establishment process using MRZ password, but afterwards a SELECT APDU command with missing cryptogram is used
Version	3.00
References	[R1] Part 11 4.4
Profile	PACE
Preconditions	1. Reset the chip 2. EF.CardAccess has been read correctly
Test scenario	1. Send the given MSE: Set AT APDU with MRZ password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <reference private key>' - <PACE OID> is a valid PACE OID fitting the implemented algorithm - The private key reference SHALL be included in the APDU if and only if the domain parameters are ambiguous, i,e more than one parameterId values are assigned to the same PACE OID in the PACEInfo entries in EF.CardAccess 2. Send the given General Authenticate APDU to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to map the nonce: '10 86 00 00 <Lc> 7C <L_{7C}> 81 <L_{81}> <mapping data> <Le>' 4. Perform key agreement: '10 86 00 00 <Lc> 7C <L_{7C}> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 7C <L_{7C}> 85 <L_{85}> <authentication token> <Le>' 6. To verify that the correct SM is required, send the given SELECT APDU for EF.CardAccess . The secure messaging is correctly computed (corrects checksum and cryptogram) but Data Object DO 87 is removed. '0C A4 02 0C <Lc> 8E 08 <Checksum> 00 with missing cryptogram (Tag 87 is removed)' 7. To verify that the error in previous step has terminated the SM session, send the given READ Binary APDU for EF.CardAccess '0C B0 (80 OR <sfi.EF.CardAccess>) 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' 2. 7C <L_{7C}> '80' <L_{80}> <encrypted nonce> '90 00' 3. 7C <L_{7C}> '82' <L_{82}> <mapping data> '90 00' 4. 7C <L_{7C}> '84' <L_{84}> <ephemeral public key> '90 00' 5. 7C <L_{7C}> '86' <L_{86}> <authentication token> '90 00' 6. ISO_checking_error or ISO_execution_error without SM 7. ISO_checking_error or ISO_execution_error without SM
Post conditions	1. None

3.15 Unit Test ISO7816_Q – Select and Read EF.CardAccess

3.15.1 Introduction

This test unit contains all mandatory tests regarding the Select and Read binary command applied to EF.CardAccess. EF.CardAccess SHALL be a transparent elementary file contained in the master file.

3.15.2 Test Case ISO7816_Q_1

Purpose	Accessing EF.CardAccess with explicit file selection and Read Binary
Version	2.0
References	[R1] Part 11 4.2
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1C' 2. Send the following Read Binary APDU to the eMRTD '00 B0 00 00 01'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return byte '31' followed by status bytes '90 00'

3.15.3 Test Case ISO7816_Q_2

Purpose	Accessing EF.CardAccess with implicit file selection (ReadBinary with SFI)
Version	2.0
References	[R1] Part 11 4.2
Profile	PACE
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B0 9C 00 01'
Expected results	1. The eMRTD SHALL return byte '31' followed by status bytes '90 00'

3.15.4 Test Case ISO7816_Q_3

Purpose	Accessing EF.CardAccess with explicit file selection and Read Binary OddIns
Version	2.0
References	[R1] Part 11 4.2
Profile	PACE, OddIns
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1C' 2. Send the following Read Binary APDU to the eMRTD '00 B1 00 00 04 54 02 00 00 03'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return bytes '53 L_{data} data' followed by status bytes '90 00'

3.15.5 Test Case ISO7816_Q_4

Purpose	Accessing EF.CardAccess with implicit file selection (ReadBinary OddIns with SFI)
Version	2.0
References	[R1] Part 11 4.2
Profile	PACE, OddIns
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B1 00 1C 04 54 02 00 00 03'

Expected results	1. The eMRTD SHALL return byte '53 L_{data} data' followed by status bytes '90 00'

3.16 Unit Test ISO7816_R – Active Authentication

3.16.1 Introduction

This test unit contains all mandatory tests regarding execution of the Active Authentication security mechanism.
In this Test unit, if the response data do not fit into a short length response APDU as defined in [R1] Part 11 6.1.4, use extended length fields in the command APDU.
<Lc> = '08' for short length fields and <Lc> = '00 00 08' for extended length fields
<Le> = '00' for short length fields and <Le> = '00 00' for extended length fields
<DO 97> = '97 01 00' for short length fields and <DO 97> = '97 02 00 00' for extended length fields.

3.16.2 void

Removed in version v3.00.

3.16.3 Test Case ISO7816_R_2

Purpose	Verify the behavior of a BAC-protected eMRTD in response to the INTERNAL AUTHENTICATE command (positive test)
Version	2.0
References	[R1] Part 11 6.1
Profile	AA, BAC
Preconditions	1. Reset the chip 2. The LDS application SHALL have been selected. 3. The BAC mechanism SHALL have been performed.
Test scenario	1. Send the given INTERNAL AUTHENTICATE valid SM command: '0C 88 00 00 <Lc> 87 <L_{87}> 01 <cryptogram> <DO 97> 8E 08 <checksum> <Le>' - <cryptogram> contains the SM encryption of the following data : '55 66 77 88 11 22 33 44'
Expected results	1. Response data and '90 00' in a valid SM response APDU.

3.16.4 Test Case ISO7816_R_3

Purpose	Verify the behavior of an eMRTD in response to the INTERNAL AUTHENTICATE command when RND.IFD < 8 bytes
Version	2.04
References	[R1] Part 11 6.1
Profile	AA
Preconditions	1. Reset the chip 2. The LDS application SHALL have been selected. 3. If any access control mechanism is supported, this mechanism SHALL have been performed.
Test scenario	1. Send the INTERNAL AUTHENTICATE with RND.IFD '11 22 33 44'. If any access control mechanism is supported, the command APDU SHALL be encoded in a valid SM format.
Expected results	1. ISO checking error or ISO execution error. If any access control mechanism is supported, the response APDU SHALL be encoded in a valid SM format

3.16.5 Test Case ISO7816_R_4

Purpose	Verify the behavior of an eMRTD in response to the INTERNAL AUTHENTICATE command when RND.IFD > 8 bytes
Version	2.04
References	[R1] Part 11 6.1
Profile	AA
Preconditions	1. Reset the chip

	2. The LDS application SHALL have been selected. 3. If any access control mechanism is supported, this mechanism SHALL have been performed.
Test scenario	1. Send the INTERNAL AUTHENTICATE with RND.IFD '11 22 33 44 55 66 77 88 99'. If any access control mechanism is supported, the command APDU SHALL be encoded in a valid SM format
Expected results	1. ISO checking error or ISO execution error. If any access control mechanism is supported, the response APDU SHALL be encoded in a valid SM format

3.16.6 Test Case ISO7816_R_5

Purpose	This test checks the RSA signature that is generated during Active Authentication.
Version	2.0
References	[R1] Part 11 6.1 RFC-5280 RFC-3279
Profile	AA, AA-RSA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD 2. EF.DG15 contains a valid RSA public key 3. The RND.IFD and the signature that has been generated by the eMRTD are available
Test scenario	1. Obtain the plaintext signature from the Internal Authenticate response. 2. Decipher the Active Authentication signature using the Public Key from EF.DG15. 3. "Signature Opening" - Check the leftmost 2 bits of the Recoverable String. 4. "Signature Opening" - Check the trailer of the Recoverable String. 5. "Intermediate String Recovery" - Retrieve the number of padding bits from the beginning of the Recoverable String. 6. "Trailer Recovery" - Check the last byte of the Recoverable String. 7. "Hash Code Checking" - Retrieve the hash code from the Recoverable String
Expected results	1. The length of the signature SHALL be in accordance with the length of the public key from EF.DG15 2. The length of the deciphered signature SHALL be in accordance with the length of the public key from EF.DG15 3. The leftmost 2 bits of the Recoverable String SHALL be equal to '01'b. 4. The rightmost 4 bits of the Recoverable String SHALL be equal to '1100'b. 5. The number of padding bits equal to '0'b following the 3rd bit of the Recoverable String SHALL be less than 8. 6. The trailer of the Recoverable String SHALL be 'BC' if option 1 is used with SHA-1 '38CC' if option 2 is used with SHA-224 '34CC' if option 2 is used with SHA-256 '36CC' if option 2 is used with SHA-384 '35CC' if option 2 is used with SHA-512 7. . The hash code SHALL match the hash calculated over M1||M2 (M1 is the nonce that has been generated by the eMRTD; M2 is RND.IFD)

3.16.7 Test Case ISO7816_R_6

Purpose	This test checks the ECDSA signature that is generated by the eMRTD during Active Authentication.
Version	2.0

References	[R1] Part 11 6.1 RFC-5280 RFC-3279
Profile	AA, AA-ECDSA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD 2. EF.DG14 has been retrieved from the eMRTD 3. EF.DG15 contains a valid EC public key 4. The RND.IFD and the signature that has been generated by the eMRTD are available
Test scenario	1. Obtain the plaintext signature from the Internal Authenticate Response. 2. Verify the signature using ECDSA with the selected hash provided in DG14.
Expected results	1. The length of the signature SHALL be in accordance with the length of the public key from EF.DG15 2. Signature verification SHALL be successful.

3.16.8 Test Case ISO7816_R_7

Purpose	Verify the behavior of a PACE-protected eMRTD in response to the INTERNAL AUTHENTICATE command (positive test)
Version	2.11
References	[R1] Part 11 6.1
Profile	AA, PACE
Preconditions	1. Reset the chip 2. The PACE mechanism SHALL have been performed. 3. The LDS application SHALL have been selected.
Test scenario	1. Send the given INTERNAL AUTHENTICATE valid SM command: '0C 88 00 00 <Lc> 87 <L_{87}> 01 <cryptogram> <DO 97> 8E 08 <checksum> <Le>' - <cryptogram> contains the SM encryption of the following data : '55 66 77 88 11 22 33 44'
Expected results	1. Response data and '90 00' in a valid SM response APDU.

3.17 Unit Test ISO7816_S – Select and Read EF.CardSecurity

3.17.1 Introduction

This test unit contains all mandatory tests regarding the Select and Read binary command applied to EF.CardSecurity. EF.CardSecurity SHALL be a transparent elementary file contained in the master file. This test unit covers also security access conditions.

3.17.2 Test Case ISO7816_S_1

Purpose	Accessing EF.CardSecurity with explicit file selection and Read Binary
Version	2.08
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	PACE, PACE-CAM or LDS2
Preconditions	1. Reset the chip 2. The PACE protocol SHALL have been performed using the MRZ-derived password.
Test scenario	1. Send the following Select APDU to the eMRTD '0C A4 02 0C <Lc> 87 <L_{87}> 01 <cryptogram> 8E 08 <checksum> 00' - <Cryptogram> contains EF.CardSecurity File Identifier '01 1D' encrypted according to the SM being used 2. Send the following Read Binary APDU to the eMRTD '0C B0 00 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return byte '30' followed by status bytes '90 00'

3.17.3 Test Case ISO7816_S_2

Purpose	Accessing EF.CardSecurity with implicit file selection (ReadBinary with SFI)
Version	2.08
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	PACE, PACE-CAM or LDS2
Preconditions	1. Reset the chip 2. The PACE protocol SHALL have been performed using the MRZ-derived password.
Test scenario	1. Send the following Read Binary APDU to the eMRTD '0C B0 9D 00 <Lc> 97 01 01 8E 08 <checksum> 00'
Expected results	1. The eMRTD SHALL return byte '30' followed by status bytes '90 00'

3.17.4 Test Case ISO7816_S_3

Purpose	Accessing EF.CardSecurity with explicit file selection and Read Binary OddIns
Version	2.08
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	(PACE, PACE-CAM or LDS2), OddIns
Preconditions	1. Reset the chip 2. The PACE protocol SHALL have been performed using the MRZ-derived password.
Test scenario	1. Send the following Select APDU to the eMRTD '0C A4 02 0C <Lc> 87 <L_{87}> 01 <cryptogram> 8E 08 <checksum> 00' - <Cryptogram> contains EF.CardSecurity File Identifier '01 1D' encrypted according to the SM being used 2. Send the following Read Binary APDU to the eMRTD '0C B1 00 00 <Lc> 85 <L_{85}> <cryptogram> 97 01 03 8E 08 <checksum> 00'

	- <Cryptogram> contains '54 02 00 00' encrypted according to the SM being used.
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return bytes '53 01 30' followed by status bytes '90 00'

3.17.5 Test Case ISO7816_S_4

Purpose	Accessing EF.CardSecurity with implicit file selection (ReadBinary OddIns with SFI)
Version	2.08
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	(PACE, PACE-CAM or LDS2), OddIns
Preconditions	1. Reset the chip 2. The PACE protocol SHALL have been performed using the MRZ-derived password.
Test scenario	1. Send the following Read Binary APDU to the eMRTD '0C B1 00 1D <Lc> 85 <L_{85}> <cryptogram> 97 01 03 8E 08 <checksum> 00' - <Cryptogram> contains '54 02 00 00' encrypted according to the SM being used.
Expected results	1. The eMRTD SHALL return byte '53 01 30' followed by status bytes '90 00'

3.17.6 Test Case ISO7816_S_5

Purpose	Accessing EF.CardSecurity with explicit file selection and Read Binary but without performing PACE
Version	3.00
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	PACE, PACE-CAM or LDS2
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1D' 2. Send the following Read Binary APDU to the eMRTD '00 B0 00 00 00'
Expected results	1. The eMRTD SHALL return status bytes '90 00' or '69 82'. If '69 82' is returned, skip the next step. 2. The eMRTD SHALL return status bytes '69 82'

3.17.7 Test Case ISO7816_S_6

Purpose	Accessing EF.CardSecurity with implicit file selection (ReadBinary with SFI) but without performing PACE
Version	3.00
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	PACE, PACE-CAM or LDS2
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B0 9D 00 01'
Expected results	1. The eMRTD SHALL return status bytes '69 82'

3.17.8 Test Case ISO7816_S_7

Purpose	Accessing EF.CardSecurity with explicit file selection and Read Binary OddIns but without performing PACE
Version	3.00
References	[R1] Part 11 9.2.11 & Part 10 3.11.4

Profile	(PACE, PACE-CAM or LDS2), OddIns
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 01 1D' 2. Send the following Read Binary APDU to the eMRTD '00 B1 00 00 04 54 02 00 00 03'
Expected results	1. The eMRTD SHALL return status bytes '90 00' or '69 82'. If '69 82' is returned, skip the next step. 2. The eMRTD SHALL return status bytes '69 82'

3.17.9 Test Case ISO7816_S_8

Purpose	Accessing EF.CardSecurity with implicit file selection (ReadBinary OddIns with SFI) but without performing PACE
Version	3.00
References	[R1] Part 11 9.2.11 & Part 10 3.11.4
Profile	(PACE, PACE-CAM or LDS2), OddIns
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B1 00 1D 04 54 02 00 00 03'
Expected results	1. The eMRTD SHALL return status bytes '69 82'

3.18 Unit Test ISO7816_T – Select and Read EF.ATR/INFO

3.18.1 Introduction

This test unit contains all mandatory tests regarding the Select and Read binary command applied to EF.ATR/INFO. EF. ATR/INFO SHALL be a transparent elementary file contained in the master file.

This test unit is applicable for LDS2 or if the cryptographic algorithms and key sizes selected by the issuing State require the use of extended length in PACE, AA, CA or TA mechanisms. This condition SHALL be verified before performing the tests.

3.18.2 Test Case ISO7816_T_1

Purpose	Accessing EF.ATR/INFO with explicit file selection and Read Binary
Version	3.00
References	[R1] Part 10 3.11
Profile	ICAO (see 3.18.1)
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 2F 01' 2. Send the following Read Binary APDU to the eMRTD '00 B0 00 00 01'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return one byte of data followed by status bytes '90 00'

3.18.3 Test Case ISO7816_T_2

Purpose	Accessing EF.ATR/INFO with implicit file selection (ReadBinary with SFI)
Version	3.00
References	[R1] Part 10 3.11
Profile	ICAO (see 3.18.1)
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B0 81 00 01'
Expected results	1. The eMRTD SHALL return one byte of data followed by status bytes '90 00'

3.18.4 Test Case ISO7816_T_3

Purpose	Accessing EF.ATR/INFO with explicit file selection and Read Binary OddIns
Version	3.00
References	[R1] Part 10 3.11
Profile	ICAO, OddIns (see 3.18.1)
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 2F 01' 2. Send the following Read Binary APDU to the eMRTD '00 B1 00 00 04 54 02 00 00 03'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return bytes '53 L_{data} data' followed by status bytes '90 00'

3.18.5 Test Case ISO7816_T_4

Purpose	Accessing EF.ATR/INFO with implicit file selection (ReadBinary OddIns with SFI)
Version	3.00
References	[R1] Part 10 3.11
Profile	ICAO, OddIns (see 3.18.1)
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B1 00 01 04 54 02 00 00 03'

Expected results	1. The eMRTD SHALL return byte '53 L_{data} data' followed by status bytes '90 00'

3.19 Unit Test ISO7816_U – Select and Read EF.DIR

3.19.1 Introduction

This test unit contains all mandatory tests regarding the Select and Read binary command applied to EF.DIR. EF.DIR SHALL be a transparent elementary file contained in the master file.

3.19.2 Test Case ISO7816_U_1

Purpose	Accessing EF.DIR with explicit file selection and Read Binary
Version	3.00
References	[R1] Part 10 3.11
Profile	LDS2
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 2F 00' 2. Send the following Read Binary APDU to the eMRTD '00 B0 00 00 01'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return byte '61' followed by status bytes '90 00'

3.19.3 Test Case ISO7816_U_2

Purpose	Accessing EF.DIR with implicit file selection (ReadBinary with SFI)
Version	3.00
References	[R1] Part 10 3.11
Profile	LDS2
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B0 9E 00 01'
Expected results	1. The eMRTD SHALL return byte '61' followed by status bytes '90 00'

3.19.4 Test Case ISO7816_U_3

Purpose	Accessing EF.DIR with explicit file selection and Read Binary OddIns
Version	3.00
References	[R1] Part 10 3.11
Profile	LDS2, OddIns
Preconditions	1. Reset the chip
Test scenario	1. Send the following Select APDU to the eMRTD '00 A4 02 0C 02 2F 00' 2. Send the following Read Binary APDU to the eMRTD '00 B1 00 00 04 54 02 00 00 03'
Expected results	1. The eMRTD SHALL return status bytes '90 00' 2. The eMRTD SHALL return bytes '53 L_{data} data' followed by status bytes '90 00'

3.19.5 Test Case ISO7816_U_4

Purpose	Accessing EF.DIR with implicit file selection (ReadBinary OddIns with SFI)
Version	3.00
References	[R1] Part 10 3.11
Profile	LDS2, OddIns
Preconditions	1. Reset the chip
Test scenario	1. Send the following Read Binary APDU to the eMRTD '00 B1 00 1E 04 54 02 00 00 03'
Expected results	1. The eMRTD SHALL return byte '53 L_{data} data' followed by status bytes '90 00'

4 Logical Data Structure Tests

4.1 Introduction

The “logical data structure” test layer analyses the encoding of the LDS objects stored on an eMRTD. This layer contains several test units, one for each LDS object (DG 1 - 16, EF.COM and EF.SOD). Another test unit verifies the integrity and consistency of the different data structures. The tests specified in this layer can be performed using a regular eMRTD or with following input data from a different source (e.g. file). The test configuration document specifies the source of the data.

4.2 Unit Test LDS_A - Tests for the EF.COM LDS Object

4.2.1 Introduction

This unit includes all test cases concerning the EF.COM element. The general LDS header encoding is tested as well as the referred LDS and Unicode version numbers. The consistency of the LDS data group list with respect to the available LDS data group objects is checked in a different test unit.

4.2.2 Test Case LDS_A_1

Purpose	This test checks the template tag; the encoded LDS element starts with.
Version	1.1
References	[R1] Part 10 5.1
Profile	ICAO
Preconditions	Encoded EF.COM object in binary format as read from the eMRTD.
Test scenario	1. Check the very first byte of the EF.COM element
Expected results	1. First byte SHALL be '60'
Postconditions	None

4.2.3 Test Case LDS_A_2

Purpose	This test checks the encoding of LDS element length.
Version	1.1
References	[R1] Part 10 5.1
Profile	ICAO
Preconditions	Encoded EF.COM object in binary format as read from the eMRTD.
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the given LDS object
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the given LDS object.
Postconditions	None

4.2.4 Test Case LDS_A_3

Purpose	This test checks the LDS version referred by the EF.COM element
Version	2.03
References	[R1] Part 10 4.5 & 5.1
Profile	ICAO
Preconditions	Encoded EF.COM object in binary format as read from the eMRTD.
Test scenario	1. Search for configured tag '5F 01' 2. Verify the length of the tag '5F 01' 3. Verify the length of LDS version DE. 4. Verify the LDS version.
Expected results	1. Tag SHALL be present. 2. The bytes that follow the tag SHALL contain a valid length encoding. 3. Length SHALL be 4.

	4. The specified LDS version SHALL be '30 31 30 37' or '30 31 30 38' and shall be coherent with the version defined in the ICS
Postconditions	None

4.2.5 Test Case LDS_A_4

Purpose	This test checks the Unicode version referred by the EF.COM element
Version	1.1
References	[R1] Part 10 5.1
Profile	ICAO
Preconditions	Encoded EF.COM object in binary format as read from the eMRTD.
Test scenario	1. Search for configured tag '5F 36' 2. Verify the length of the tag '5F 36' 3. Verify the length of the Unicode version DE. 4. Verify the Unicode version.
Expected results	1. Tag SHALL be present. 2. The bytes that follow the tag SHALL contain a valid length encoding. 3. The length SHALL be 6. 4. The specified Unicode version SHALL be '30 34 30 30 30 30'.
Postconditions	None

4.2.6 Test Case LDS_A_5

Purpose	This test checks the list of the present LDS data groups
Version	1.1
References	[R1] Part 10 5.1
Profile	ICAO
Preconditions	Encoded EF.COM object in binary format as read from the eMRTD.
Test scenario	1. Search for configured tag '5C' 2. Verify the length of the tag '5C' 3. Verify if mandatory LDS data groups are present. 4. Verify the validity of present LDS data groups.
Expected results	1. Tag SHALL be present. 2. The bytes that follow the tag SHALL contain a valid length encoding 3. The list SHALL at least contain the tags for the mandatory LDS data groups '61', '75'. 4. The list SHALL contain only valid LDS data group tags as specified in [R1], i.e. '61', '75', '63', '76', '65', '66', '67', '68', '69', '6A', '6B', '6C', '6D', '6E', '6F', '70'
Postconditions	None

4.3 Unit Test LDS_B - Tests for the DataGroup 1 LDS object

4.3.1 Introduction

This unit includes all test cases concerning the DG 1 element (MRZ). The general LDS header encoding is tested as well as the format of the MRZ and the calculation of the check digits.

Unit test B uses the following definitions in accordance with [R1]:

- A denotes the set of ASCII encoded alphabetic characters {“A”, “B”, … , “Z”}
- N denotes the set of ASCII encoded numeric characters {“0”, “1”, … , “9”}
- S denotes the set of ASCII encoded special characters {“<”}

4.3.2 Test Case LDS_B_1

Purpose	This test verifies the template tag with which the encoded LDS element starts.
Version	1.1
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD.
Test scenario	1. Check the very first byte of the EF.DG1 element
Expected results	1. First byte SHALL be '61'
Postconditions	None

4.3.3 Test Case LDS_B_2

Purpose	This test verifies the encoding of LDS element length.
Version	1.1
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD.
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the given LDS object
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the given LDS object.
Postconditions	None

4.3.4 Test Case LDS_B_3

Purpose	This test verifies the encoding of the MRZ data object.
Version	1.1
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD.
Test scenario	1. Verify the length of the tag '5F 1F' 2. Verify that the length encoding is correct. 3. Verify that the encoded length equals the remaining size of DG1.
Expected results	1. The first bytes of the LDS element data SHALL be the tag for the MRZ data object. 2. The bytes that follow the MRZ data object tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 3. The encoded length SHALL match the remaining size of the given DG1 object.
Postconditions	None

4.3.5 Test Case LDS_B_4

Purpose	This test checks the format of the document code.

Version	3.00
References	[R1] Part 5 4.2.2.1 &Part 6 4.2.2.1 [R1] Part 10 4.7.1 & Part 4 4.2.2.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. For all documents, the format of the document type is composed of the bytes 1 and 2 of the MRZ.
Test scenario	1. Analyze the first two characters of the MRZ (document code).
Expected results	1. If TD3, the first characters of document code shall be 'P' and the second character shall be an element of A, S. If TD2, the first characters of document code shall be 'A', 'C' or 'I' and the second character shall be an element of A, N or S. 'V' shall not be used, and 'C' shall not be used after 'A'. If TD1, the first characters of document code shall be 'A', 'C' or 'I' and the second character shall be an element of A, N or S. 'V' shall not be used, 'I' shall not be used after 'A' and 'C' shall not be used after 'A'.
Postconditions	None

4.3.6 Test Case LDS_B_5

Purpose	This test checks the format of the issuing state of the MRZ.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. For all documents, the format of the issuing state is composed of bytes 3 to 5 (inclusive) of the MRZ.
Test scenario	1. Analyze the next three characters of the MRZ (issuing state).
Expected results	1. The characters of the issuing state SHALL be elements of A that MAY be followed by elements of S.
Postconditions	None

4.3.7 Test Case LDS_B_6

Purpose	This test verifies the format of the holder name of the MRZ.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the holder name of the MRZ as described below: If TD3, the format of the holder name is composed of bytes 6 to 44. If TD2, the format of the holder name is composed of bytes 6 to 36. If TD1, the format of the holder name is composed of bytes 61 to 90.
Test scenario	1. Analyze the characters of the MRZ (holder name).
Expected results	1. The characters of the holder name SHALL be elements of A or S. The holder name SHALL start with a character that is an element of A.
Postconditions	None

4.3.8 Test Case LDS_B_7

Purpose	This test verifies the format of the document number.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the Document Number and its check digit as described below: If TD3, the format of the document number is composed of bytes 45 to 53. The check digit is the next character (54).

	If TD2, the format of the document number is composed of bytes 37 to 45. The check digit is the next character (46). If the check digit is an element of S, then the remaining characters of the document number are composed of bytes 65 until an element of S is encountered (71 or before), at which point the document number ends at 2 character positions before (69 or before) and the check digit is the next character (70 or before). If TD1, the format of the document number is composed of bytes 6 to 14. The check digit is the next character (15). If the check digit is an element of S, then the remaining characters of the document number are composed of bytes 16 until an element of S is encountered (30 or before), at which point the document number ends at 2 character positions before (28 or before) and the check digit is the next character (29 or before).
Test scenario	1. Analyze the characters of the MRZ (document number). 2. Analyze the next character of the MRZ (check digit).
Expected results	1. The characters of the document number SHALL be elements of A or N that MAY be followed by elements of S. 2. The document number check digit must be an element of N and SHALL be correct.
Postconditions	None

4.3.9 Test Case LDS_B_8

Purpose	This test verifies the format of the nationality.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the nationality as described below: If TD3, the format of the nationality is composed of bytes 55 to 57. If TD2, the format of the nationality is composed of bytes 47 to 49. If TD1, the format of the nationality is composed of bytes 46 to 48.
Test scenario	1. Analyze the three characters of the MRZ (nationality).
Expected results	1. The characters of the nationality SHALL be elements of A that MAY be followed by elements of S.
Postconditions	None

4.3.10 Test Case LDS_B_9

Purpose	This test verifies the format of the date of birth.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the date of birth and its check digit as described below: If TD3, the format of the date of birth and its check digit is composed of bytes 58 to 64. If TD2, the format of the date of birth and its check digit is composed of bytes 50 to 56. If TD1, the format of the date of birth and its check digit is composed of bytes 31 to 37.
Test scenario	1. Analyze the 6 characters of the MRZ (date of birth). 2. Analyze the next character of the MRZ (check digit).
Expected results	1. The six characters SHALL be elements of N or S. The format is YYMMDD, where MM SHALL be an element of {01 to 12} or S; and DD SHALL be an element of {01 to 31} or S.

	2. The check digit of the date of birth SHALL be an element of N and SHALL be correct. For data elements in which the number does not occupy all available character positions, the symbol < shall be used to complete vacant positions and shall be given the value of zero for the purpose of calculating the check digit.
Postconditions	None

4.3.11 Test Case LDS_B_10

Purpose	This test verifies the format of the sex.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the format of the sex. If TD3, the format of the sex is the byte 65. If TD2, the format of the sex is byte 57. If TD1, the format of the sex is byte 38.
Test scenario	1. Analyze the character of the MRZ (sex).
Expected results	1. The character of the sex SHALL be an element of {“F”, “M”, “<”}.
Postconditions	None

4.3.12 Test Case LDS_B_11

Purpose	This test verifies the format of the date of expiry.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the date of expiry and its check digit as described below: If TD3, the format of the date of expiry and its check digit is composed of bytes 66 to 72. If TD2, the format of the date of birth and its check digit is composed of bytes 58 to 64. If TD1, the format of the date of birth and its check digit is composed of bytes 39 to 45.
Test scenario	1. Analyze the 6 characters of the MRZ (date of expiry). 2. Analyze the next character of the MRZ (check digit).
Expected results	1. The six characters SHALL be elements of N. The format is YYMMDD, where MM SHALL be an element of {01 to 12}; and DD SHALL be an element of {01 to 31}. 2. The check digit of the date of expiry SHALL be an element of N and SHALL be valid.
Postconditions	None

4.3.13 Test Case LDS_B_12

Purpose	This test verifies the format of the optional data.
Version	2.02
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the optional data and check digit for TD2 as described below: If TD3, the format of the optional data and its check digit is composed of bytes 73 to 87. If TD2, the format of the optional data is composed of bytes 65 to 71 (no check digit for TD2).

	If TD1, the format of the optional data is composed of bytes 16 to 30, and from 49 to 59 (no check digit for TD1).
Test scenario	1. Analyze the characters of the MRZ (optional data). 2. Analyze the next character of the MRZ (check digit).
Expected results	1. The characters of the optional data SHALL be elements of A, N or S. 2. If the optional data's check digit is not present, skip this step. If the optional data is not empty (i.e. partly or wholly composed of elements of A, N), the optional data's check digit SHALL be element of N and SHALL be correct. Else, the optional data's check digit SHALL be an element of {“0” or “<”}.
Postconditions	None

4.3.14 Test Case LDS_B_13

Purpose	This test verifies the format of composite check digit.
Version	2.0
References	[R1] Part 10 4.7.1.1
Profile	ICAO
Preconditions	Encoded EF.DG1 object in binary format as read from the eMRTD. Extract the check digit as described below: If TD3, the format of the composite check digit is the byte 88. If TD2, the format of the composite check digit is the byte 72. If TD1, the format of the composite check digit is the byte 60.
Test scenario	1. Analyze the character of the MRZ (composite check digit). If TD3, the check digit is calculated by concatenating bytes 45 to 54, 58 to 64, and 66 to 87. If TD2, the check digit is calculated by concatenating bytes 37 to 46, 50 to 56, and 58 to 71. If TD1, the check digit is calculated by concatenating bytes 6 to 37, 39 to 45, and 49 to 59.
Expected results	1. The character of the composite check digit SHALL be an element of N and it SHALL be correct.
Postconditions	None

4.4 Unit Test LDS_C - Tests for the DataGroup 2 LDS object

4.4.1 Introduction

This unit includes all test cases concerning the DG 2 element (Face). The general LDS header encoding is tested as well as the CBEFF encoded biometric template and ISO 19794 coding [R4] of the biometric object itself. Since the CBEFF and the ISO specification allow a very high degree of freedom, this unit contains tests for the mandatory elements as specified in the LDS.

Some additional (optional) tests verify the encoding optional elements. The general rule for this optional test is that if an optional element is present, it SHALL be encoded according to the corresponding specification otherwise the test fails.

4.4.2 Test Case LDS_C_1

Purpose	This test checks the template tag; the encoded DataGroup 2 element starts with.
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the very first byte of the EF.DG2 element
Expected results	1. First byte SHALL be '75'
Postconditions	None

4.4.3 Test Case LDS_C_2

Purpose	This test checks the encoding of LDS element length.
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the given LDS object
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the given LDS object.
Postconditions	None

4.4.4 Test Case LDS_C_3

Purpose	This test checks the encoding of the Biometric Information Group Template.
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the first tag in the DG 2 data. 2. Verify the length of the DG 2 data. 3. Verify that the encoded length is less than size of DG 2.
Expected results	1. Tag SHALL be '7F 61'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL NOT exceed the remaining bytes of the DG 2 data element.
Postconditions	None

4.4.5 Test Case LDS_C_4

Purpose	This test checks the encoding of the number of instances stored in the Biometric Information Group Template.

Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the first tag inside the group template 2. Verify the length of the “number of instances” data object. 3. Check the number of instances.
Expected results	1. Tag SHALL be '02'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The number of instances SHALL be 1.
Postconditions	None

4.4.6 Test Case LDS_C_5

Purpose	This test checks the encoding of the first biometric information template.
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the tag of the biometric information template. 2. Verify the length of the “biometric information template” data object. 3. Verify that the encoded length is less than rest of size of DG 2.
Expected results	1. Tag SHALL be '7F 60'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL NOT exceed the remaining bytes of the DG 2 element.
Postconditions	None

4.4.7 Test Case LDS_C_6

Purpose	This test checks the encoding of the biometric header template tag.
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the presence of the biometric header template tag with the configured tag. 2. Verify the length of the “biometric header template” data object. 3. Verify that the encoded length is less than rest of size of DG 2.
Expected results	1. Tag SHALL be 'A1'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL NOT exceed the remaining bytes of the DG 2 element.
Postconditions	None

4.4.8 Test Case LDS_C_7

Purpose	This test checks the presence/encoding of the CBEFF element "format owner".
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO

Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD. The tested CBEFF element is part of biometric header template located in LDS_C_06.
Test scenario	1. Check the presence of the “format owner” tag. 2. Verify the length of the “format owner” data object. 3. Check the length of the “format owner” value. 4. Verify the “format owner” value.
Expected results	1. Tag SHALL be present and be '87'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 2 bytes. 4. The value of the format owner SHALL be a registered CBEFF owner. It SHALL be '01 01'.
Postconditions	None

4.4.9 Test Case LDS_C_8

Purpose	This test checks the presence/encoding of the CBEFF element "format type".
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD. The tested CBEFF element is part of biometric header template located in LDS_C_06.
Test scenario	1. Check the presence of the format type tag. 2. Verify the length of the “format type” data object. 3. Check the length of the “format type” value. 4. Verify the “format type” value.
Expected results	1. Tag SHALL be present and be '88'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 2 bytes. 4. The value of the format type SHALL be a registered CBEFF type. It SHALL be '00 08'.
Postconditions	None

4.4.10 Test Case LDS_C_9

Purpose	This test checks the encoding of the biometric data object tag.
Version	1.1
References	[R1] Part 10 6.2
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD. The biometric data object is part of the biometric information template tested in LDS_C_05.
Test scenario	1. Check the presence of the biometric data object tag. 2. Verify the length of the biometric data object. 3. Verify that the encoded length is less than rest of size of DG 2.
Expected results	1. Tag SHALL be present and be '5F 2E' 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL NOT exceed the remaining bytes of the DG 2 element.
Postconditions	None

4.4.11 Test Case LDS_C_10

Purpose	This test checks the encoding of the facial header block.
Version	1.1
References	[R1] Part 10 6.2 [R4]
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD. The biometric data object is part of the biometric data object tested in LDS_C_09.
Test scenario	1. Check the first 4 bytes of the header block (Format identifier) 2. Check the next 4 bytes of the header block (Version number) 3. Check the record length element. 4. Check the Number of Facial Images element.
Expected results	1. The format identifier SHALL be '46 41 43 00'. 2. The version number SHALL be '30 31 30 00'. 3. The length SHALL NOT exceed the remaining bytes of the DG2 element and SHALL match the encoded length of the biometric data object. 4. The number of facial images SHALL at least be 1.
Postconditions	None

4.4.12 Test Case LDS_C_11

Purpose	This test checks the encoding of the facial information block. This test is mandatory for the first facial information block and SHOULD be repeated for further optional facial images.
Version	1.1
References	[R1] Part 10 6.2 [R4]
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the Facial Record Data Length. 2. Check the number of facial feature points. 3. Check the gender element. 4. Check the eye color element. 5. Check the hair color element. 6. Check Pose Angle - Yaw. 7. Check Pose Angle - Pitch. 8. Check Pose Angle - Roll. 9. Check Pose Angle Uncertainty –Yaw. 10. Check Pose Angle Uncertainty –Pitch. 11. Check Pose Angle Uncertainty –Roll.
Expected results	1. The Facial Record Data Length SHALL be at least 32 bytes and SHALL NOT exceed the remaining size of the biometric data object. 2. The size of the feature point structures (8 * number of facial feature points) SHALL NOT exceed the remaining size of the biometric data object 3. The gender SHALL be encoded as '00', '01', '02', or 'FF'. 4. The eye color SHALL be encoded as '00', '01', '02', '03', '04', '05', '06', '07', or 'FF'. 5. The hair color SHALL be encoded as '00', '01', '02', '03', '04', '05', '06', '07', or 'FF'. 6. The Pose Angle - Yaw SHALL be equal or less than 181. 7. The Pose Angle - Pitch SHALL be equal or less than 181.

	8. The Pose Angle - Roll SHALL be equal or less than 181. 9. The Pose Angle Uncertainty - Yaw SHALL be equal or less than 181. 10. The Pose Angle Uncertainty - Pitch SHALL be equal or less than 181. 11. The Pose Angle Uncertainty - Roll SHALL be equal or less than 181.
Postconditions	None

4.4.13 Test Case LDS_C_12

Purpose	This test checks the encoding of the facial feature points. It is conditional and applies only if there are feature points encoded. This test SHOULD be repeated for every present feature point. See LDS_C_11 for the number of feature points.
Version	1.1
References	[R1] Part 10 6.2 [R4]
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the feature point type.
Expected results	1. The feature point type SHALL be 1.
Postconditions	None

4.4.14 Test Case LDS_C_13

Purpose	This test checks the encoding of the image information block. This test is mandatory for the first image information block and SHOULD be repeated for further optional facial images.
Version	1.1
References	[R1] Part 10 6.2 [R4]
Profile	ICAO
Preconditions	Encoded EF.DG2 object in binary format as read from the eMRTD.
Test scenario	1. Check the face image type. 2. Check the image data type.
Expected results	1. The face image type SHALL be encoded as '00', '01', or '02'. 2. The image data type SHALL be encoded as '00' or '01'.
Postconditions	None

4.5 Unit Test LDS_D - Tests for the SOD LDS object

4.5.1 Introduction

This unit includes all test cases concerning the EF.SOD element. The general LDS header encoding is tested as well as the contained CMS (PKCS#7) signed content object.

In order verify the signing certificate signature the corresponding country signing certificate is needed. For the verification of the LDS security object, the binary LDS data group objects and the EF.COM is needed as read from the eMRTD.

4.5.2 Test Case LDS_D_1

Purpose	This test checks the template tag; the encoded DataGroup 2 element starts with.
Version	1.1
References	[R1] Part 10 5.2
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD.
Test scenario	1. Check the very first byte of the EF.SOD element.
Expected results	1. First byte SHALL be '77'.
Postconditions	None

4.5.3 Test Case LDS_D_2

Purpose	This test checks the encoding of LDS element length.
Version	1.1
References	[R1] Part 10 5.2
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD.
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the given LDS object
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the given LDS object.
Postconditions	None

4.5.4 Test Case LDS_D_3

Purpose	This test checks the ASN#1 encoding of a PCKS#7 signedData object.
Version	1.1
References	[R1] Part 10 5.2 & Part 12 7
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD.
Test scenario	1. Check that the element has a sound ASN.1 structure.
Expected results	1. The PKCS#7 signed data object included as the value in the LDS true template SHALL be encoded according to the DER format. Note that the use of indefinite length is not authorized.
Postconditions	None

4.5.5 Test Case LDS_D_4

Purpose	This test checks the value that is encoded into the signedData element.
Version	2.11
References	[R1] Part 10 5.2 & Part 12
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD.
Test scenario	1. Check the SignedData version value. 2. Check the digestAlgorithms list. 3. Check the eContentType.

	4. Get the LDS Security Object version and check the certificates list.
Expected results	1. The version number SHALL be 3. 2. All OIDs SHALL be valid. This list SHOULD contain all used digestAlgorithms in this signedData container. It SHALL contain only digestAlgorithms specified in[R1] Part 12 4.1.6.4: 2.16.840.1.101.3.4.2.1 (SHA-256) 2.16.840.1.101.3.4.2.2 (SHA-384) 2.16.840.1.101.3.4.2.3 (SHA-512) 2.16.840.1.101.3.4.2.4 (SHA-224) 3. The eContentType SHALL have OID id-icao-mrtd-security-ldsSecurityObject. 4. If LDS Security Object version is 0, the certificate list MAY contain the Document Signer Certificate. If LDS Security Object version is 1, the certificate list SHALL contain the Document Signer Certificate.
Postconditions	None

4.5.6 Test Case LDS_D_5

Purpose	This test checks the SignerInfo element of the signedData structure. The signedData Structure SHALL at least contain one signer info. If there is more than one signer info, although this is not recommended, this test SHALL be repeated for each element.
Version	2.08
References	[R1] Part 10 5.2 & Part 12
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD.
Test scenario	1. Check the signer info version value. 2. Check the choice of the sid element. 3. Check if the certificate identified in the sid is included in the signed data certificates list or available in the PKD. 4. Check the digestAlgorithm identifier. 5. Check the signedAttrs element. 6. Check the MessageDigest Attribute. 7. Check the SigningTime attribute if present. 8. Check the signatureAlgorithm element. 9. Check the signature element. It is verified with the signer certificates public key and the hash value produced over the signedAttributes.
Expected results	1. The version number SHALL be 1 or 3. 2. The choice of the sid element SHALL match the signer info version value. (Version 1 if issuerandSerialNumber is used and 3 if subjectKeyIdentifier is used). 3. Certificate SHALL be available. 4. The digestAlgorithmID SHALL refer to an algorithm specified in[R1] Part 12 4.1.6.4. 5. The signed attributes list SHALL contain the MessageDigest attribute. 6. The value of the message digest attribute SHALL match the hash value of the eContent element. (Using the digestAlgorithm specified above) 7. If there's a SigningTime attribute present, the signing time SHALL be within the validity period of the signing certificate. 8. The signature algorithm SHALL refer to an algorithm specified in [R1] Part 12 4.4. 9. The signature SHALL be valid.

Postconditions	None

4.5.7 Test Case LDS_D_6

Purpose	This test checks the LDS Security Object stored as eContent in the signedData Object. The LDS Security Object is stored as the eContent element in the signedData Structure.
Version	2.08
References	[R1] Part 10 5.2 & Part 12
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD. For the LDS data group hash verification this test needs also the binary LDS data group objects as read from the eMRTD.
Test scenario	1. Check the ASN.1 encoding of the LDS Security Object. 2. Check the security object version element. 3. Check the digestAlgorithm identifier. 4. Check the DataGroupHash Sequence. 5. Check the dataGroup numbers in the DataGroup Hash Sequence. 6. Check the dataGroup numbers in the DataGroup Hash Sequence. 7. Check the dataGroup hash values in the Hash Sequence. Compare the hash value with the corresponding LDS data group binary objects. 8. If the security object version is 1, check LDS Version Info element. If the security object version is 0, check LDS Version Info element does not exist.
Expected results	1. The object SHALL be encoded according to the DER syntax. Note that the use of indefinite length is not authorized. 2. The version number SHALL be 0 or 1. If version is 0, LDS Version defined in the ICS shall be 01.07 If version is 1, LDS Version defined in the ICS shall be 01.08 3. The digestAlgorithmID SHALL refer to an algorithm specified in [R1] Part 12 4.1.6.4. 4. The Sequence SHALL contain at least 2 entries for DG 1 and 2. 5. The Sequence SHALL contain a hash value for all present LDS data groups. There SHALL be no additional hash value for non-existing LDS data groups. 6. The referred dataGroups SHALL match the DataGroup list in the EF.COM. 7. All hash values SHALL be valid. 8. If the security object version is 1, LDS Version Info element shall be present and shall be '30 31 30 38'. If the security object version is 0, the LDS Version Info element is absent.
Postconditions	None

4.5.8 Test Case LDS_D_7

Purpose	This test checks the signing certificate used to verify the EF.SOD object. The certificate can be read from the SOD object or SHALL be retrieved from the PKD.
Version	2.08
References	[R1] Part 10 5.2 & Part 12
Profile	ICAO
Preconditions	Encoded EF.SOD object in binary format as read from the eMRTD. For the verification of the signing certificate signature, the country signing certificate is required.
Test scenario	1. Check the ASN.1 encoding of the signing certificate. 2. Check the signing certificate version element.

	3. Check the signature element. 4. Check the certificates validity period element. 5. Check the certificates issuer element. 6. Check the subjectPublicKeyInfo element. 7. Check the AuthorityKeyIdentifier extension in the signing certificate. 8. Check that the SubjectKeyIdentifier extension of the country signing certificate matches the AuthorityKeyIdentifier of the signing certificate. 9. Check the keyUsage extension of the signing certificate. 10. Check the signatureAlgorithm element. 11. Verify the signatureValue of the signing certificate with the public key of the country signing certificate.
Expected results	1. The object SHALL be encoded according to the DER syntax. Note that the use of indefinite length is not authorized. 2. The version SHALL be v3 (Value for v3 is 2). 3. The algorithm specified here SHALL match the OID in the signatureAlgorithm field. 4. It SHALL use UTC time until 2049 and from then on GeneralizedTime. The validity period of the signing certificate SHALL be within the validity period of the country signing certificate. Note: It is not necessary that the certificate is still valid; it SHALL only have been valid at signing time, which is tested in LDS_D_5. 5. The issuer SHALL match the subject of the provided country signing certificate. 6. This element SHALL refer to an algorithm specified in [R1] Part 12 4.1.6. 7. This extension SHALL be present and SHALL contain a keyIdentifier value. 8. AuthorityKeyIdentifier SHALL match the SubjectKeyIdentifier of the country signing certificate. 9. The keyUsage extension SHALL be “critical” and the digitalSignature bit SHALL be asserted. 10. The signatureAlgorithm element SHALL refer to an algorithm specified in [R1] Part 12 4.4. 11. The certificate signature SHALL be valid.
Postconditions	None

4.6 Unit Test LDS_E – Tests for the DataGroup 14 LDS object

4.6.1 Introduction

This unit contains all mandatory tests regarding the coding of LDS data group 14. DG14 contains SecurityInfo structures related to the various security protocols supported by the eMRTD.
For the PACE profile, DG14 SHALL contain a copy of each of the SecurityInfos stored in EF.CardAccess.
If DG14 contains multiple instances of the same element type (ChipAuthentication, ChipAuthenticationPublicKeyInfo, TerminalAuthenticationInfo), the corresponding test cases have to be performed for each element. A test case is only rated as a PASS if all passes of a test case are performed without any failure, so that all test runs for one test case lead to a single result.

4.6.2 Test Case LDS_E_1

Purpose	Test the LDS tag of the data group 14 object
Version	2.11
References	[R1] Part 10 6.14
Profile	PACE or CA or AA, AA-ECDSA or TA
Preconditions	1. Data group 14 SHALL have been read from the eMRTD
Test scenario	1. Verify the hex value of the very first byte of the data group 14 content. It SHALL contain the LDS tag for this data group. 2. The tag is followed by an ASN.1 style encoded length of the data group 14 object. This length SHALL be encoded correctly according to the ASN.1 specification. 3. The encoded length SHALL NOT exceed the overall length of the read to group object.
Expected results	1. '6E' 2. true 3. true

4.6.3 Test Case LDS_E_2

Purpose	Test the ASN.1 encoding of the SecurityInfos for Chip Authentication
Version	2.11
References	[R1] Part 11 9.2, 9.2.5, 9.2.6
Profile	CA
Preconditions	1. Data group 14 SHALL have been read from the eMRTD
Test scenario	1. The data content of the data group 14 SHALL be encoded according to the SecurityInfos syntax definition. 2. The SecurityInfos set SHALL contain at least one ChipAuthenticationPublicKeyInfo element with one of the protocol OID defined in [R1] Part 11 9.2.6 (id-PK-DH or id-PK-ECDH). The test LDS_E_3 SHALL be performed for each ChipAuthenticationPublicKeyInfo element which has such an OID. 3. If at least one ChipAuthenticationInfo element (with OID a. id-CA-DH-3DES-CBC-CBC or b. id-CA-DH-AES-CBC-CMAC-128 or c. id-CA-DH-AES-CBC-CMAC-192 or d. id-CA-DH-AES-CBC-CMAC-256 or e. id-CA-ECDH-3DES-CBC-CBC or f. id-CA-ECDH-AES-CBC-CMAC-128 or g. id-CA-ECDH-AES-CBC-CMAC-192 or h. id-CA-ECDH-AES-CBC-CMAC-256)

	is present, there SHALL be at least one ChipAuthenticationInfo element with the version element set to 1.
Expected results	1. true 2. true 3. true

4.6.4 Test Case LDS_E_3

Purpose	Test the ASN.1 encoding of the ChipAuthenticationPublicKeyInfo
Version	2.11
References	[R1] Part 11 9.1, 9.2.6
Profile	CA
Preconditions	1. Data group 14 SHALL have been read from the eMRTD 2. The data group 14 is parsed and this test is repeated for each ChipAuthenticationPublicKeyInfo element containing the OID (id-PK-DH or id-PK-ECDH) as defined in [R1] Part 11 9.2.6
Test scenario	1. The ChipAuthenticationPublicKeyInfo element must follow the ASN.1 syntax definition in [R1] Part 11 9.2.6. 2. The presence of the key reference in the ChipAuthenticationPublicKeyInfo SHALL be coherent with the ICS 3. The algorithm identifier SHALL match to the Key agreement protocol and be one of the following: • DHKeyAgreement (OID: 1.2.840.113549.1.3.1) • ecPublicKey (OID: 1.2.840.10045.2.1) 4. The parameters SHALL follow PKCS #3 (DH) or KAEG specification (ECDH). For DH verify that • $0 < g < p$, that is both should be positive and g should be less than p. • If private value length l is present, verify that $l > 0$ and $2^{l-1} < p$. In case of ECDH verify that • prime $p > 2$ • curve parameter $0 \leq a < p$ • curve parameter $0 \leq b < p$ • $4a^3 + 27b^2 \neq 0$ • base point G is on the curve, with both coordinates in range $0 \ldots p - 1$ • Cofactor $f > 0$ • order r of base point $r > 0$, $r \neq p$ • $r * f \leq 2p$ 5. The public key value SHALL follow PKCS#3 (DH) or [R5] specification (ECDH) For DH verify that • $0 < y < p$ For ECDH verify that public point Y is on the curve, with both coordinates in range $0 \ldots p - 1$
Expected results	1. true 2. true 3. true

	4. true 5. true

4.6.5 Test Case LDS_E_4

Purpose	Test the ASN.1 encoding of the ChipAuthenticationInfo
Version	2.11
References	[R1] Part 11 9.2.5
Profile	CA
Preconditions	1. Data group 14 SHALL have been read from the eMRTD 2. The data group 14 is parsed and this test is repeated for each ChipAuthenticationInfo element containing one of the OIDs defined in [R1] Part 11 9.2.5 (see LDS_E_2) and the version element set to 1.
Test scenario	1. The ChipAuthenticationInfo element must follow the ASN.1 syntax definition [R1] Part 11 9.2.5. 2. The presence of the key reference in the ChipAuthenticationInfo SHALL be coherent with the ICS 3. If the key reference is present in the ChipAuthenticationInfo element, there SHALL be also on ChipAuthenticationPublicKeyInfo element with this key reference.
Expected results	1. true 2. true 3. true

4.6.6 Test Case LDS_E_5

Purpose	Test the coherency between the DG14 and EF.CardAccess
Version	2.0
References	[R1] Part 11 9.2.8
Profile	PACE
Preconditions	1. Data group 14 SHALL have been read from the eMRTD
Test scenario	1. Check the SecurityInfo structures stored in the CardAccess are duplicated in the DG14.
Expected results	1. Each SecurityInfo structure stored in the CardAccess file is also present in DG14.

4.6.7 Void

Removed in version 3.00.

4.6.8 Test Case LDS_E_7

Purpose	Test the ASN.1 encoding of the ActiveAuthenticationInfo
Version	2.0
References	[R1] Part 11 9.2.4
Profile	AA, AA-ECDSA
Preconditions	1. EF.DG14 has been retrieved from the eMRTD 2. EF.DG15 has been retrieved from the eMRTD
Test scenario	1. Check the SecurityInfos element 2. Check the version of the ActiveAuthenticationInfo 3. Check the signatureAlgorithm of the ActiveAuthenticationInfo 4. Check the hash algorithm output length of the signatureAlgorithm
Expected results	1. The SecurityInfos SHALL contain an ActiveAuthenticationInfo element with protocol identifier id-icao-mrtd-security-aaProtocolObject : (OID : 2.23.136.1.1.5). 2. The version SHALL be encoded as INTEGER and SHALL be 1

	3. The signatureAlgorithm SHALL be one of the following : - ecdsa-plain-SHA224: (OID: 0.4.0.127.0.7.1.1.4.1.2) - ecdsa-plain-SHA256: (OID: 0.4.0.127.0.7.1.1.4.1.3) - ecdsa-plain-SHA384: (OID: 0.4.0.127.0.7.1.1.4.1.4) - ecdsa-plain-SHA512: (OID: 0.4.0.127.0.7.1.1.4.1.5) 4. The Hash algorithm output length is same length or shorter than the length of the ECDSA key in use.

4.6.9 Test Case LDS_E_8

Purpose	Test that EF.DG14 contains at least one valid set of SecurityInfos for Chip Authentication. A chip supporting PACE-CAM must also support CA.
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE, PACE-CAM
Preconditions	1. Data group 14 SHALL have been read from the eMRTD
Test scenario	1. Check the SecurityInfo structures stored in the DG14
Expected results	1. At least one valid set of SecurityInfos for Chip Authentication SHALL be present

4.6.10 Test Case LDS_E_9

Purpose	Test the ASN.1 encoding of the TerminalAuthentication
Version	3.00
References	[R1] Part 11 9.2.8
Profile	TA
Preconditions	1. Data group 14 SHALL have been read from the eMRTD
Test scenario	1. Check the SecurityInfos element. 2. Check the version of the TerminalAuthenticationInfo
Expected results	1. The SecurityInfos SHALL contain at least one TerminalAuthenticationInfo element with the general protocol object identifier id-TA only: OID: 0.4.0.127.0.7.2.2.2. No TA protocol like id-TA-RSA or id-TA-ECDSA shall be present. 2. The version SHALL be encoded as INTEGER and SHALL be 1.

4.6.11 Test Case LDS_E_10

Purpose	Test the ASN.1 encoding of the TerminalAuthentication
Version	3.00
References	[R1] Part 11 9.2.8
Profile	TA, TA-EU
Preconditions	1. Data group 14 SHALL have been read from the eMRTD
Test scenario	1. If present, check the efCVCA of the TerminalAuthenticationInfo
Expected results	1. If the efCVCA element is present, it shall contain the fid element encoded as an OCTET STRING (SIZE (2)) If sfid is present, it shall be encoded as an OCTET STRING (SIZE (1))

4.7 Unit Test LDS_F - Tests for the EF.CVCA Object

4.7.1 Introduction

This unit covers all tests about the coding of the EF.CVCA file containing the trust point for the certificate verification process.

4.7.2 Test case LDS_F_1

Purpose	Positive test - Test the content of the EF.CVCA file
Version	3.00
References	[R1] Part 11 Annex K.2
Profile	TA, TA-EU
Preconditions	1. The EF.CVCA SHALL have been read from the eMRTD
Test scenario	1. The size of the EF.CVCA file SHALL be exactly 36 bytes. 2. The EF.CVCA file SHALL contain at least one at most two Certificate Authority Reference objects. 3. Each object SHALL start with the tag '42' 4. The encoded object length of each object SHALL NOT exceed 16 bytes. 5. Any remaining bytes of the EF.CVCA content SHALL be padded with '00'.
Expected results	1. true 2. true 3. true 4. true 5. true
Post conditions	1. None

4.8 Unit Test LDS_G - Tests for the EF.DG3 LDS Object

4.8.1 Introduction

This unit includes all test cases concerning the DG 3 element (fingerprint). As the purpose of this test specification is not to test biometrics, only the general DG3 header (as defined in [ICAO 9303]) is tested to ensure minimum conformance.

4.8.2 Test case LDS_G_1

Purpose	Positive test - This test checks the template tag; the encoded DG 3 element starts with.
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD.
Test scenario	1. Check the very first byte of the EF.DG3 element
Expected results	1. First byte SHALL be '63'
Post conditions	1. None

4.8.3 Test case LDS_G_2

Purpose	Positive test - This test checks the encoding of DG3 length bytes.
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD.
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the DG3
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the DG3 value bytes.
Post conditions	1. None

4.8.4 Test case LDS_G_3

Purpose	Positive test - This test checks the encoding of the Biometric Information Group Template (BIGT).
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD.
Test scenario	1. Check that the first tag in the DG 3 value is the BIGT tag. 2. Verify the length of the BIGT. 3. Verify that the BIGT is the only information in the DG3.
Expected results	1. Tag SHALL be '7F 61'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the number of remaining bytes of the DG 3 data element, except if the BIGT contains no BITs (no fingerprints). In this case the BIGT MAY be followed by a DO 53 containing random data to prevent the static hash value.
Post conditions	1. None

4.8.5 Test case LDS_G_4

Purpose	Positive test - This test checks the encoding of the number of instances stored in the Biometric Information Group Template (BIGT).
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3

Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD.
Test scenario	1. Check the first tag inside the BIGT. 2. Verify the length of the “number of instances” data object. 3. Verify the value of the “number of instances” data object.
Expected results	1. Tag SHALL be '02'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The number of instances SHALL match the actual number of encoded Biometric Information Templates (tag 7F 60).
Post conditions	1. None

4.8.6 Test case LDS_G_5

Purpose	Positive test - This test checks the encoding of the Biometric Information Template (BIT).
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the tag of the BIT. 2. Verify the length of the BIT data object. 3. Verify that the encoded length matches the size of the BIT.
Expected results	1. Tag SHALL be '7F 60'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the effective length of the encoded BIT
Post conditions	1. None

4.8.7 Test case LDS_G_6

Purpose	Positive test - This test checks the encoding of the Biometric Header Template (BHT).
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the tag of the BHT 2. Verify the length of the BHT data object. 3. Verify that the encoded length matches the size of the BHT.
Expected results	1. Tag SHALL be 'A1'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the effective length of the encoded BHT
Post conditions	1. None

4.8.8 Test case LDS_G_7

Purpose	Positive test - This test checks the presence/encoding of the CBEFF element "format owner".
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the “format owner” tag. 2. Verify the length of the “format owner” data object. 3. Check the length of the “format owner” value.

	4. Verify the “format owner” value.
Expected results	1. Tag SHALL be present and be '87'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 2 bytes. 4. The value of the format owner SHALL be a registered CBEFF owner. It SHALL be '01 01' for the first instance of BIT. All registered format owner can be found at www.ibia.org.
Post conditions	1. None

4.8.9 Test case LDS_G_8

Purpose	Positive test - This test checks the presence/encoding of the CBEFF element "format type".
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the “format type” tag. 2. Verify the length of the “format type” data object. 3. Check the length of the “format type” value. 4. Verify the “format type” value.
Expected results	1. Tag SHALL be present and be '88'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 2 bytes. 4. The value of the format type SHALL be a registered CBEFF type. It SHALL be '0007' for the first instance of BIT. All registered format types can be found at www.ibia.org.
Post conditions	1. None

4.8.10 Test case LDS_G_9

Purpose	Positive test - This test checks the presence/encoding of the CBEFF element “ biometric subtype”.
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the “biometric subtype” tag. 2. Verify the length of the “biometric subtype” data object. 3. Check the length of the “biometric subtype” value. 4. Verify the “biometric subtype” value.
Expected results	1. Tag SHALL be present and be '82'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 1 byte. 4. The value of the format type SHALL be a registered CBEFF biometric subtype. The values for the biometric subtype are defined in ISO 19785-3.
Post conditions	1. None

4.8.11 Test case LDS_G_10

Purpose	Positive test - This test checks the encoding of the Biometric Data Block (BDB) tag.
Version	3.00
References	[R1] Part 10 4.7.3

Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the BDB tag. 2. Verify the length of the BDB. 3. Verify that the encoded length match the size of encoded BDB
Expected results	1. Tag SHALL be present and be '5F 2E' 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the effective length of the encoded BDB
Post conditions	1. None

4.8.12 Test case LDS_G_11

Purpose	Positive test - This test verifies the consistency between the CBEFF format type and the BDB format identifier of the BIT
Version	3.00
References	[R1] Part 10 4.7.3
Profile	DG3
Preconditions	1. Encoded EF.DG3 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the first four bytes of the BDB
Expected results	1. The value SHALL be '46 49 52 00' ('F' 'I' 'R' 0x00).
Post conditions	1. None

4.9 Unit Test LDS_H - Tests for the EF.DG4 LDS Object

4.9.1 Introduction

This unit includes all test cases concerning the DG 4 element (Iris). As the purpose of this test specification is not to test biometrics, only the general DG4 header (as defined in [ICAO 9303]) is tested to ensure minimum conformance.

4.9.2 Test case LDS_H_1

Purpose	Positive test - This test checks the template tag; the encoded DG 4 element starts with.
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD.
Test scenario	1. Check the very first byte of the EF.DG4 element
Expected results	1. First byte SHALL be '76'
Post conditions	1. None

4.9.3 Test case LDS_H_2

Purpose	Positive test - This test checks the encoding of DG4 length bytes.
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD.
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the DG4
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the DG4 value bytes.
Post conditions	1. None

4.9.4 Test case LDS_H_3

Purpose	Positive test - This test checks the encoding of the Biometric Information Group Template (BIGT).
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD.
Test scenario	1. Check that the first tag in the DG4 value is the BIGT tag. 2. Verify the length of the BIGT. 3. Verify that the BIGT is the only information in the DG4.
Expected results	1. Tag SHALL be '7F 61'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the number of remaining bytes of the DG 4 data element, except if the BIGT contains no BITs (no iris images). In this case the BIGT MAY be followed by a DO 53 containing random data to prevent the static hash value.
Post conditions	1. None

4.9.5 Test case LDS_H_4

Purpose	Positive test - This test checks the encoding of the number of instances stored in the Biometric Information Group Template (BIGT).
Version	3.00
References	[R1] Part 10 4.7.4

Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD.
Test scenario	1. Check the first tag inside the BIGT. 2. Verify the length of the “number of instances” data object. 3. Verify the value of the “number of instances” data object.
Expected results	1. Tag SHALL be '02'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The number of instances SHALL match the actual number of encoded Biometric Information Templates (tag 7F 60).
Post conditions	1. None

4.9.6 Test case LDS_H_5

Purpose	Positive test - This test checks the encoding of the Biometric Information Template (BIT).
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the tag of the BIT. 2. Verify the length of the BIT data object. 3. Verify that the encoded length match the size of the BIT.
Expected results	1. Tag SHALL be '7F 60'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the effective length of the encoded BIT
Post conditions	1. None

4.9.7 Test case LDS_H_6

Purpose	Positive test - This test checks the encoding of the Biometric Header Template (BHT).
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the tag of the BHT 2. Verify the length of the BHT data object. 3. Verify that the encoded length match the size of the BHT.
Expected results	1. Tag SHALL be present and be 'A1'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the effective length of the encoded BHT
Post conditions	1. None

4.9.8 Test case LDS_H_7

Purpose	Positive test - This test checks the presence/encoding of the CBEFF element "format owner".
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the “format owner” tag. 2. Verify the length of the “format owner” data object.

	3. Check the length of the “format owner” value. 4. Verify the “format owner” value.
Expected results	1. Tag SHALL be present and be '87'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 2 bytes. 4. The value of the format owner SHALL be a registered CBEFF owner. It SHALL be '01 01' for the first instance of BIT. All registered format owner can be found at www.ibia.org.
Post conditions	1. None

4.9.9 Test case LDS_H_8

Purpose	Positive test - This test checks the presence/encoding of the CBEFF element "format type".
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the “format type” tag. 2. Verify the length of the “format type” data object. 3. Check the length of the “format type” value. 4. Verify the “format type” value.
Expected results	1. Tag SHALL be present and be '88'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 2 bytes. 4. The value of the format type SHALL be a registered CBEFF type. It SHALL be "0009' or '000B' for the first instance of BIT. All registered format types can be found at www.ibia.org.
Post conditions	1. None

4.9.10 Test case LDS_H_9

Purpose	Positive test - This test checks the presence/encoding of the CBEFF element " biometric subtype".
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the “biometric subtype” tag. 2. Verify the length of the “biometric subtype” data object. 3. Check the length of the “biometric subtype” value. 4. Verify the “biometric subtype” value.
Expected results	1. Tag SHALL be present and be '82'. 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The length of the value field SHALL be 1 byte. 4. The value of the format type SHALL be a registered CBEFF biometric subtype. The values for the biometric subtype are defined in ISO 19785-3.
Post conditions	1. None

4.9.11 Test case LDS_H_10

Purpose	Positive test - This test checks the encoding of the Biometric Data Block (BDB) tag.
Version	3.00

References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the presence of the BDB tag. 2. Verify the length of the BDB. 3. Verify that the encoded length matches the size of encoded BDB
Expected results	1. Tag SHALL be present and be '5F 2E' 2. This element SHALL have a valid encoded length (According to ASN.1 encoding rules). 3. The encoded length SHALL match the effective length of the encoded BDB
Post conditions	1. None

4.9.12 Test case LDS_H_11

Purpose	Positive test - This test verifies the consistency between the CBEFF format type and the BDB format identifier of the BIT
Version	3.00
References	[R1] Part 10 4.7.4
Profile	DG4
Preconditions	1. Encoded EF.DG4 object in binary format as read from the eMRTD. 2. This test SHALL be repeated for each instance of BIT
Test scenario	1. Check the first four bytes of the BDB
Expected results	1. The value SHALL be '49 49 52 00' ('I' 'I' 'R' 0x00).
Post conditions	1. None

4.10 Unit Test LDS_I – Tests for the EF.CardAccess

4.10.1 Introduction

This unit contains all mandatory tests regarding the presence and coding of PACE-related SecurityInfo structures in EF.CardAccess.
If the EF.CardAccess contains multiple instances of the same element type (PACEInfo, PACEDomainParameterInfo), the corresponding test cases have to be performed for each element. A test case is only rated as a PASS if all passes of a test case are performed without any failure, so that all test runs for one test case lead to a single result.

4.10.2 Test Case LDS_I_1

Purpose	Test the ASN.1 encoding of the SecurityInfos
Version	2.0
References	[R1] Part 11 9
Profile	PACE
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD
Test scenario	1. Check the EF.CardAccess file data 2. Check the SecurityInfos.
Expected results	1. The data content of the EF.CardAccess SHALL be encoded according to the SecurityInfos syntax definition 2. - At least one PACEInfo using a standardized domain parameter SHALL be present. - SecurityInfos may contain additional entries indicating support for other protocols. The inspection system may discard any unknown entry. - The data structure PACEDomainParameterInfo is REQUIRED if the eMRTD chip provides proprietary domain parameters for PACE, and SHALL be omitted otherwise

4.10.3 Test Case LDS_I_2

Purpose	Test the ASN.1 encoding of the PACEInfo
Version	2.08
References	[R1] Part 11 9
Profile	PACE
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. The Card Access content is parsed and this test is repeated for each PACEInfo element containing an OID as defined in [R1] .
Test scenario	1. The PACEInfo element must follow the ASN.1 syntax definition as defined in [R1]. 2. The algorithm identifier SHALL be one of the following: – id-PACE-DH-GM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.1.1) – id-PACE-DH-GM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.1.2) – id-PACE-DH-GM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.1.3) – id-PACE-DH-GM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.1.4) – id-PACE-ECDH-GM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.2.1) – id-PACE-ECDH-GM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.2.2) – id-PACE-ECDH-GM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.2.3)

	– id-PACE-ECDH-GM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.2.4) – id-PACE-DH-IM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.3.1) – id-PACE-DH-IM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.3.2) – id-PACE-DH-IM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.3.3) – id-PACE-DH-IM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.3.4) – id-PACE-ECDH-IM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.4.1) – id-PACE-ECDH-IM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.4.2) – id-PACE-ECDH-IM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.4.3) – id-PACE-ECDH-IM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.4.4) – id-PACE-ECDH-CAM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.6.2) – id-PACE-ECDH-CAM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.6.3) – id-PACE-ECDH-CAM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.6.4) 3. Check that a valid OID is present for each declared configuration in table 1 4. The version SHALL be encoded as INTEGER and SHALL be 2 5. Check the ParameterId matches the domain parameters: The ParameterId if present SHALL be one of the following (see table 9 of [R1] Part 11) – '00' if 1024-bit MODP Group with 160-bit Prime Order Subgroup is used. – '01' if 2048-bit MODP Group with 224-bit Prime Order Subgroup is used. – '02' if 2048-bit MODP Group with 256-bit Prime Order Subgroup is used. – '08' if NIST P-192 is used – '09' if BrainpoolP192r1 is used – '10' if NIST P-224 is used – '11' if BrainpoolP224r1 is used – '12' if NIST P-256 is used – '13' if BrainpoolP256r1 is used – '14' if BrainpoolP320r1 is used – '15' if NIST P-384 is used – '16' if BrainpoolP384r1 is used – '17' if BrainpoolP512r1 is used – '18' if NIST P-521 is used – Above '32' (included) for proprietary domain parameters
Expected results	1. true 2. true 3. true 4. true 5. true

4.10.4 Test Case LDS_I_3

Purpose	Test the ASN.1 encoding of the PACEDomainParameterInfo
Version	2.08
References	[R1] Part 11 9
Profile	PACE
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. The Card Access content is parsed and this test is repeated for each proprietary PACEDomainParameterInfo element. The test is skipped if no PACEDomainParameterInfo object is found.
Test scenario	1. The PACEDomainParameterInfo element must follow the ASN.1 syntax definition 2. The protocol identifier SHALL be one of the following: – id-PACE-DH-GM (OID : 0.4.0.127.0.7.2.2.4.1) – id-PACE-ECDH-GM (OID : 0.4.0.127.0.7.2.2.4.2) – id-PACE-DH-IM (OID : 0.4.0.127.0.7.2.2.4.3) – id-PACE-ECDH-IM (OID : 0.4.0.127.0.7.2.2.4.4) – id-PACE-ECDH-CAM (OID : 0.4.0.127.0.7.2.2.4.6) 3. The algorithm identifier SHALL match to the key agreement protocol and be one of the following: – dhpublicnumber (OID: 1.2.840.10046.2.1) – ecPublicKey (OID: 1.2.840.10045.2.1) 4. The parameters SHALL follow PKCS #3 (DH) or KAEG specification (ECDH). For DH verify that – $0 < g < p$, that is both should be positive and g should be less than p. – If private value length l is present, verify that $l > 0$ and $2^{l-1} < p$. In case of ECDH verify that – prime $p > 2$ – curve parameter $0 \leq a < p$ – curve parameter $0 \leq b < p$ – $4a^3 + 27b^2 \neq 0$ – base point G is on the curve, with both coordinates in range $0 \ldots p - 1$ – Cofactor $f > 0$ – order r of base point $r > 0$, $r \neq p$ – $r * f \leq 2p$ – the generator point is encoded in uncompressed format according to [R5], i.e. '04||x||y' 5. If a ParameterId is present in the PACEDomainParameterInfo element, there SHALL be at least one PACEInfo element with this ParameterId. 6. If a ParameterId is present in the PACEDomainParameterInfo element, it must be larger than 31.
Expected results	1. true 2. true 3. true

	4. true 5. true 6. true

4.10.5 Test Case LDS_I_4

Purpose	Verify that EF.CardAccess contains at least one valid PACEInfo for PACE-GM or PACE-IM as an additional mapping procedure if PACE-CAM is supported
Version	2.08
References	[R1] Part 11 4.4
Profile	PACE, PACE-CAM
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD
Test scenario	1. At least one of the following algorithm identifier SHALL be present in a valid PACEInfo: – id-PACE-DH-GM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.1.1) – id-PACE-DH-GM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.1.2) – id-PACE-DH-GM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.1.3) – id-PACE-DH-GM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.1.4) – id-PACE-ECDH-GM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.2.1) – id-PACE-ECDH-GM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.2.2) – id-PACE-ECDH-GM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.2.3) – id-PACE-ECDH-GM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.2.4) – id-PACE-DH-IM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.3.1) – id-PACE-DH-IM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.3.2) – id-PACE-DH-IM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.3.3) – id-PACE-DH-IM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.3.4) – id-PACE-ECDH-IM-3DES-CBC-CBC (OID : 0.4.0.127.0.7.2.2.4.4.1) – id-PACE-ECDH-IM-AES-CBC-CMAC-128 (OID : 0.4.0.127.0.7.2.2.4.4.2) – id-PACE-ECDH-IM-AES-CBC-CMAC-192 (OID : 0.4.0.127.0.7.2.2.4.4.3) – id-PACE-ECDH-IM-AES-CBC-CMAC-256 (OID : 0.4.0.127.0.7.2.2.4.4.4)
Expected results	1. true

4.11 Unit Test LDS_J – Tests for the DataGroup 15 LDS object

4.11.1 Introduction

This unit contains all mandatory tests regarding the coding of LDS data group 15. DG15 contains the public key information required for the Active Authentication mechanism.

4.11.2 Test Case LDS_J_1

Purpose	This test checks the template tag that the encoded EF.DG15 element starts with.
Version	2.0
References	[R1] Part 10 6.15
Profile	AA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD
Test scenario	1. Check the very first byte of the EF.DG15 element
Expected results	1. First byte SHALL be '6F'

4.11.3 Test Case LDS_J_2

Purpose	This test checks the encoding of EF.DG15 element length.
Version	2.0
References	[R1] Part 10 6.15
Profile	AA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD
Test scenario	1. Analyze the encoding of the bytes that follow the template tag 2. Verify the length of the EF.DG15 object
Expected results	1. The bytes that follow the template tag SHALL contain a valid length encoding (According to ASN.1 encoding rules). 2. The encoded length SHALL match the size of the given EF.DG15 object.

4.11.4 Test Case LDS_J_3

Purpose	This test checks the DER-TLV encoding of the "Subject Public Key Info" present in EF.DG15.
Version	2.0
References	[R1] Part 11 6.1.5 RFC-5280
Profile	AA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD 2. EF.DG15 contains a SubjectPublicKeyInfo value under the '6F' tag
Test scenario	1. Search for the AA Public Key Info (Tag '30') inside EF.DG15. 2. Check the DER-TLV encoding of the AA Public Key Info 3. Check the value of the encoded AA Public Key Info
Expected results	1. Tag '30' SHALL be present. 2. The AA Public Key Info SHALL be DER-encoded. Note that the use of indefinite length is not authorized. 3. The AA Public Key Info SHALL follow the encoding of the Subject Public Key Info specified in RFC-3280.

4.11.5 Test Case LDS_J_4

Purpose	This test checks that the algorithm indicated for the Public Key in EF.DG15 is one of the algorithms specified in [R1].
Version	2.0
References	[R1] Part 11 6.1 RFC-3279
Profile	AA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD 2. EF.DG15 contains a SubjectPublicKeyInfo value under the '6F' tag 3. The SubjectPublicKeyInfo holds a sequence TLV
Test scenario	1. Search for the Algorithm Identifier (Tag '30') inside the AA Public Key Info. 2. Check the DER-TLV encoding of the Algorithm Identifier 3. Check the value of the Algorithm Identifier 4. Check the value of the algorithm indicated in the Algorithm Identifier
Expected results	1. Tag '30' SHALL be present and SHALL occur only once. 2. The Algorithm Identifier SHALL be DER-encoded. Note that the use of indefinite length is not authorized. 3. The Algorithm Identifier SHALL follow the ASN.1 encoding specified in RFC-5280. 4. The Public Key Algorithm indicated in the Algorithm Identifier SHALL be one of the algorithms indicated in RFC-3279 (i.e. the OID of the algorithm SHALL be rsaEncryption or id-ecPublicKey).

4.11.6 Test Case LDS_J_5

Purpose	This test checks the encoding of the Subject Public Key in the AA Public Key Info in EF.DG15.
Version	2.0
References	[R1] Part 11 6.1 RFC-3279 [R5] 3.2.1
Profile	AA
Preconditions	1. EF.DG15 has been retrieved from the eMRTD 2. EF.DG15 contains a SubjectPublicKeyInfo value under the '6F' tag 3. The SubjectPublicKeyInfo holds a sequence TLV
Test scenario	1. Search for the Subject Public Key (Tag '03') inside the AA Public Key Info. 2. Check the DER-TLV encoding of the Subject Public Key 3. Check that the data bits from the bit string code a valid Public Key for the algorithm indicated in the Subject Public Key Info data element. For ECDSA, check the encoding format of the public point. 4. If RSA public key supported, check that the length of the encoded RSA Public Key meets the minimum size recommendation.
Expected results	1. Tag '03' SHALL be present and SHALL occur only once. 2. The Subject Public Key SHALL be encoded as a bit-string. Note that the use of indefinite length is not authorized. 3. The data bits from the bit string SHALL code a valid Public Key for the algorithm indicated in the Subject Public Key Info data element. For ECDSA, the Uncompressed format for prime curve shall be used. 4. An RSA Public Key SHALL have a length of at least 1024 bits.

4.12 Unit Test LDS_K – Tests for the EF.CardSecurity

4.12.1 Introduction

This unit contains all mandatory tests regarding the presence and coding of SecurityInfo structures in EF.CardSecurity.
If the EF.CardSecurity contains multiple instances of the same element type ChipAuthenticationPublicKeyInfo, the corresponding test cases have to be performed for each element. A test case is only rated as a PASS if all passes of a test case are performed without any failure, so that all test runs for one test case lead to a single result.

4.12.2 Test Case LDS_K_1

Purpose	Test the ASN.1 encoding of the SecurityInfos
Version	2.08
References	[R1] Part 11 9
Profile	PACE, PACE-CAM
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. The PACE protocol SHALL have been performed using the MRZ-derived password and PACE-CAM OID. 3. EF.CardSecurity SHALL have been read from the eMRTD
Test scenario	1. The data content of the EF.CardSecurity SHALL be encoded according to the SecurityInfos syntax definition 2. There must be at least one SecurityInfo containing ChipAuthenticationPublicKeyInfo as required for PACE-CAM, with the following protocol OID : - id-PK-ECDH (OID : 0.4.0.127.0.7.2.2.1.2)
Expected results	1. true 2. true

4.12.3 Test Case LDS_K_2

Purpose	Verify the ASN.1 encoding of the chipAuthenticationPublicKey
Version	2.08
References	[R1] Part 11 9.2.6
Profile	PACE, PACE-CAM
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. The PACE protocol SHALL have been performed using the MRZ-derived password and PACE-CAM OID. 3. EF.CardSecurity SHALL have been read from the eMRTD and chipAuthenticationPublicKey is extracted from ChipAuthenticationPublicKeyInfo
Test scenario	1. The algorithm identifier SHALL match to the key agreement protocol and be the following: – ecPublicKey (OID: 1.2.840.10045.2.1) – StandardizedDomainParameter (OID: 0.4.0.127.0.7.1.2) 2. In case of ecPublicKey, the parameters SHALL follow KAEG specification (ECDH): – prime $p > 2$ – curve parameter $0 \leq a < p$ – curve parameter $0 \leq b < p$ – $4a^3 + 27b^2 \neq 0$ – base point G is on the curve, with both coordinates in range $0 \ldots p - 1$

	– Cofactor $f > 0$ – order r of base point $r > 0$, $r \neq p$ – $r * f \leq 2p$ – the generator point is encoded in uncompressed format according to [R5], i.e. '04||x||y' In case of StandardizedDomainParameter, the Parameters SHALL be one of the following (see table 9 of [R1] Part 11) – '08' if NIST P-192 is used – '09' if BrainpoolP192r1 is used – '10' if NIST P-224 is used – '11' if BrainpoolP224r1 is used – '12' if NIST P-256 is used – '13' if BrainpoolP256r1 is used – '14' if BrainpoolP320r1 is used – '15' if NIST P-384 is used – '16' if BrainpoolP384r1 is used – '17' if BrainpoolP512r1 is used – '18' if NIST P-521 is used
Expected results	1. True 2. True

4.12.4 Test Case LDS_K_3

Purpose	Test the coherency between the EF.CardSecurity and EF.CardAccess
Version	2.08
References	[R1] Part 11 9.2.8
Profile	PACE, PACE-CAM
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. The PACE protocol SHALL have been performed using the MRZ-derived password and PACE-CAM OID. 3. EF.CardSecurity SHALL have been read from the eMRTD
Test scenario	1. Check the SecurityInfo structures stored in the CardAccess are duplicated in the EF.CardSecurity.
Expected results	1. Each SecurityInfo structure stored in the CardAccess file is also present in EF.CardSecurity.

4.12.5 Test Case LDS_K_4

Purpose	Verify that the parameterID also denotes the ID of the Chip Authentication key used, i.e. the chip SHALL provide a ChipAuthenticationPublicKeyInfo with keyID equal to parameterID.
Version	2.08
References	[R1] Part 11 9
Profile	PACE, PACE-CAM
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. The PACE protocol SHALL have been performed using the MRZ-derived password and PACE-CAM OID. 3. EF.CardSecurity SHALL have been read from the eMRTD
Test scenario	1. PACEInfo for PACE-CAM contains a parameterID 2. The parameterID in PACEInfo for PACE-CAM denotes the ID of the Chip Authentication key used.

	The KeyID contained in ChipAuthenticationPublicKeyInfo in EF.CardSecurity is equal to parameterID of PACEInfo
Expected results	1. true 2. true

4.12.6 Test Case LDS_K_5

Purpose	This test checks the ASN#1 encoding of a PCKS#7 signedData object.
Version	2.11
References	[R1] Part 10 5.4
Profile	PACE, PACE-CAM
Preconditions	Encoded EF.CardSecurity object in binary format as read from the eMRTD.
Test scenario	1. Check that the ContentInfo structure has content type id-signedData and content of type SignedData.
Expected results	1. True
Postconditions	None

4.12.7 Test Case LDS_K_6

Purpose	This test checks the value that is encoded into the signedData element.
Version	2.11
References	[R1] Part 10 5.4
Profile	PACE, PACE-CAM
Preconditions	Encoded EF.CardSecurity object in binary format as read from the eMRTD.
Test scenario	1. Check the SignedData version value. 2. Check the digestAlgorithms list. 3. Check the eContentType. 4. Check the certificates list.
Expected results	1. The version number SHALL be 3. 2. All OIDs SHALL be valid. This list SHOULD contain all used digestAlgorithms in this signedData container. It SHALL contain only digestAlgorithms specified in[R1] Part 12 4.1.6.4: 2.16.840.1.101.3.4.2.1 (SHA-256) 2.16.840.1.101.3.4.2.2 (SHA-384) 2.16.840.1.101.3.4.2.3 (SHA-512) 2.16.840.1.101.3.4.2.4 (SHA-224) 3. The eContentType SHALL have OID id-SecurityObject. 4. The certificate list SHALL contain the Document Signer Certificate.
Postconditions	None

4.12.8 Test Case LDS_K_7

Purpose	This test checks the SignerInfo element of the signedData structure.
Version	2.11
References	[R1] Part 10 5.4
Profile	PACE, PACE-CAM
Preconditions	Encoded EF.CardSecurity object in binary format as read from the eMRTD.
Test scenario	1. Check the signer info version value. 2. Check the choice of the sid element. 3. Check if the certificate identified in the sid is included in the signed data certificates list. 4. Check the digestAlgorithm identifier. 5. Check the signedAttrs element. 6. Check the MessageDigest Attribute.

	7. Check the SigningTime attribute if present. 8. Check the signatureAlgorithm element. 9. Check the signature element. It is verified with the signer certificates public key and the hash value produced over the signedAttributes.
Expected results	1. The version number SHALL be 1 or 3. 2. The choice of the sid element SHALL match the signer info version value. (Version 1 if issuerandSerialNumber is used and 3 if subjectKeyIdentifier is used). 3. Certificate SHALL be available. 4. The digestAlgorithmID SHALL refer to an algorithm specified in[R1] Part 12 4.1.6. 5. The signed attributes list SHALL contain the MessageDigest attribute. 6. The value of the message digest attribute SHALL match the hash value of the eContent element. (Using the digestAlgorithm specified above) 7. If there's a SigningTime attribute present, the signing time SHALL be within the validity period of the signing certificate. 8. The signature algorithm SHALL refer to an algorithm specified in [R1] Part 12 4.1.6. 9. The signature SHALL be valid.
Postconditions	None

4.12.9 Test Case LDS_K_8

Purpose	This test checks the signing certificate used to verify the EF.CardSecurity object. The certificate SHALL be read from the CardSecurity object.
Version	2.11
References	[R1] Part 10 5.4
Profile	PACE, PACE-CAM
Preconditions	Encoded EF.CardSecurity object in binary format as read from the eMRTD. For the verification of the signing certificate signature, the country signing certificate is required.
Test scenario	1. Check the ASN.1 encoding of the signing certificate. 2. Check the signing certificate version element. 3. Check the signature element. 4. Check the certificates validity period element. 5. Check the certificates issuer element. 6. Check the subjectPublicKeyInfo element. 7. Check the AuthorityKeyIdentifier extension in the signing certificate. 8. Check that the SubjectKeyIdentifier extension of the country signing certificate matches the AuthorityKeyIdentifier of the signing certificate. 9. Check the keyUsage extension of the signing certificate. 10. Check the signatureAlgorithm element. 11. Verify the signatureValue of the signing certificate with the public key of the country signing certificate.
Expected results	1. The object SHALL be encoded according to the DER syntax. Note that the use of indefinite length is not authorized. 2. The version SHALL be v3 (Value for v3 is 2). 3. The algorithm specified here SHALL match the OID in the signatureAlgorithm field. 4. It SHALL use UTC time until 2049 and from then on GeneralizedTime. The validity period of the signing certificate SHALL be within the validity period of the country signing certificate.

	5. The issuer SHALL match the subject of the provided country signing certificate. 6. This element SHALL refer to an algorithm specified in [R1] Part 12 4.1.6. 7. This extension SHALL be present and SHALL contain a keyIdentifier value. 8. AuthorityKeyIdentifier SHALL match the SubjectKeyIdentifier of the country signing certificate. 9. The keyUsage extension SHALL be “critical” and the digitalSignature bit SHALL be asserted. 10. The signatureAlgorithm element SHALL refer to an algorithm specified in [R1] Part 12 4.1.6. 11. The certificate signature SHALL be valid.
Postconditions	None

4.12.10 Test Case LDS_K_9

Purpose	Test the coherency between the EF.CardSecurity and EF.DIR
Version	3.00
References	[R1] Part 11 9.2.10
Profile	LDS2
Preconditions	1. EF.CardAccess SHALL have been read from the eMRTD 2. EF.DIR SHALL have been read from the eMRTD 3. The PACE protocol SHALL have been performed using the MRZ-derived password. 4. EF.CardSecurity SHALL have been read from the eMRTD
Test scenario	1. The data content of the EF.CardSecurity SHALL be encoded according to the SecurityInfos syntax definition 2. There must be at least one SecurityInfo containing EFDIRInfo with the following protocol OID: – id-EFDIR (OID : 2.23.136.1.1.13) 3. check coherency between the element efDIR and the content of EF.DIR
Expected results	1. True 2. SecurityInfo containing EFDIRInfo with the following protocol OID is present: – id-EFDIR (OID : 2.23.136.1.1.13) 3. the OCTET STRING element efDIR is present and encapsulates the full copy of the content of EF.DIR

4.13 Unit Test LDS_L – Tests for the EF.ATR/INFO

4.13.1 Introduction

This unit includes all test cases concerning the EF.ATR/INFO. This file is used to store Extended Length parameters and DO for card capabilities.

4.13.2 Test case LDS_L_1

Purpose	This test checks that the EF.ATR/INFO is present if required
Version	3.00
References	[R1] Part 10 3.11.1
Profile	ICAO
Preconditions	This test is applicable only if if the cryptographic algorithms and key sizes selected by the issuing State require the use of extended length in PACE, AA, CA or TA mechanisms. This precondition SHALL be verified before performing the test. Encoded EF.ATR/INFO object in binary format as read from the eMRTD.
Test scenario	1. Verify the TLV structure of each data object listed in this file (tag, valid length, data). 2. Verify that there is a TLV object with tag '7F 66'. 3. Verify that the TLV object with tag '7F 66' contains two positive integers in a valid TLV structure.
Expected results	1. All TLV objects must be valid. 2. True. 3. This data object contains two positive integers.

4.13.3 Test case LDS_L_2

Purpose	This test checks that the DO for card capabilities encodes support of chaining in software function table.
Version	3.00
References	[R1] Part 10 3.11.1
Profile	LDS2
Preconditions	Encoded EF.ATR/INFO object in binary format as read from the eMRTD.
Test scenario	1. Verify the TLV structure of each data object listed in this file (tag, valid length, data). 2. Verify that there is a TLV object with tag '47'. 3. Verify the length of the tag '47' 4. Verify the bits b8 and b7 of the byte 3 – third software function
Expected results	1. All TLV objects must be valid. 2. True. 3. The length is 3 bytes. 4. The bits b8 and b7 are set to 1

4.13.4 Test case LDS_L_3

Purpose	This test checks that the EF.ATR/INFO is present if required
Version	3.00
References	[R1] Part 10 3.11.1
Profile	LDS2
Preconditions	Encoded EF.ATR/INFO object in binary format as read from the eMRTD.
Test scenario	1. Verify that there is a TLV object with tag '7F 66'. 2. Verify that the TLV object with tag '7F 66' contains two positive integers in a valid TLV structure.
Expected results	1. True. 2. This data object contains two positive integers which SHALL be at least 1 000 (decimal)

4.14 Unit Test LDS_M – Tests for the EF.DIR

4.14.1 Introduction

This unit includes all test cases concerning the EF.DIR. This file is used to indicate a list of applications supported by the eMRTD.

4.14.2 Test case LDS_M_1

Purpose	This test checks that the EF.DIR is present if required
Version	3.00
References	[R1] Part 10 3.11.2
Profile	LDS2 or EFDIR
Preconditions	Encoded EF.DIR object in binary format as read from the eMRTD.
Test scenario	1. Verify the TLV structure of each data object listed in this file (tag, valid length, data). 2. Verify that there are only TLV objects with tag '61'. 3. Verify that each TLV objects with tag '61' contains one TLV with tag '4F' 4. Verify that the LDS1 eMRTD Application International is present
Expected results	1. All TLV objects must be valid. 2. True. 3. True 4. A TLV with Tag '4F' and Value 'A0000002471001' shall be present.