Federal Office
for Information Security

Technical Guideline TR-03105

Conformity Tests for Official Electronic ID Documents

Part 3.3: Test Plan for eID-Cards with Advanced Security Mechanisms – EAC 2

Version 1.1
December 17th 2018

Version history

Version	Date	Editor	Description
0.30	12-10-2007	BSI/Networkers AG	EAC 2.0 conformity tests Proposal for harmonized document Working Draft
0.40	26-10-2007	BSI/Networkers AG	Editorial changes Additional test cases for layer 7
0.50	03-12-2007	BSI/Networkers AG	Editorial changes New test case definitions Including Comments from EAC 1.1 specification
0.60	18-01-2008	BSI/Networkers AG	Including changes from EAC 2.0 Public Beta 3
0.70	24-01-2008	BSI/Networkers AG	New test unit structure
0.80	23-10-2008	BSI/secunet AG	Including changes from EAC 1.12 and EAC 2.0 Almost Released
0.90	31-10-2008	BSI/secunet AG	Including changes from EAC 2.0 (Final)
1.00 beta 1	19-12-2008	BSI/secunet AG	Minor editorial changes
1.00 beta 2	20-02-2009	BSI/secunet AG	Resolved comments from DIF
1.00 beta 3	06-04-2009	BSI/secunet AG	Resolved comments from DIF
1.00 RC	06-12-2009	BSI/secunet AG	Resolved comments from DIF Including changes from EAC 2.02
1.00	24-02-2010	BSI/secunet AG	Resolved comments from DIF
1.01	2010-03-23	BSI	Minor editorial changes
1.02	2010-08-02	BSI	Correction in ISO chaining (PACE: General Authenticate command) Updated test case EAC2_DATA_A_3 for standardized domain parameters
1.03	2010-08-20	BSI	Test cases for layer 7 are updated for using standardized domain parameters

			Minor updates on test cases EAC2_ISO7816_ I_8, EAC2_ISO7816_ L_15, EAC2_ISO7816_ L_16 , EAC2_ISO7816_ L_19 Minor updates on Certificates 9:10 AT_CERT_21 and AT_CERT_22
1.1 RC1	2018-01-19	BSI/secunet AG	Integrated Amendment Release 3 Resolved comments Editorial changes Added new test suites for Manage Channel, Compare, Envelope/Get Response and CAv3
1.1 RC2	2018-04-30	BSI/secunet AG	Resolved comments of DIF AG Chip
1.1 RC3	2018-09-17	BSI/secunet AG	Resolved comments of DIF AG Chip
1.1 RC4	2018-11-22	BSI/secunet AG	Resolved comments of DIF AG Chip Handling of chips without CA-Infos in EF.CardAccess
1.1	2018-12-14	BSI	Resolution DIF AG Chip comments on RC 4

Content

1 Introduction

The TR 03105 defines a RF protocol and application test standard for eID-Cards. Version 2.0 of that document includes security mechanisms for ePassport, eID and eSign applications.

This document describes the test plan for machine-readable travel documents (eMRTDs) with advanced security mechanisms used for ePassport, eID and eSign applications referring to EAC version 2 and the corresponding dependencies.

As already known by the EAC version 1 test plan, this specification has a layer based structure. The layers 1 - 4 refer the RF protocol according to the ISO/IEC 14443 1-4 standard. Since the defined security mechanisms have no direct influence on this abstraction layer, this amendment does not contain any tests for these layers.

However, this document concentrates on the tests for the layer 6 (ISO/IEC 7816) and 7 (data group encoding).

This document is heavily based on the AFNOR/BSI test plan for EAC-passports. Especially tests for Chip and Terminal Authentication as well as the certificate structure are adopted by that document.

1.1 Abbreviations

Abbreviation	
ADH	Asynchronous Diffie-Hellman
AT	Authentication Template
BAC	Basic Access Control
CA	Chip Authentication (in MRTD security mechanism contexts)
	Certificate Authority (in certificate contexts)
CAN	Card Access Number
CAR	Certification Authority Reference
CHAT	Certificate Holder Authorization Template
CHR	Certificate Holder Reference
CSCA	Country Signing Certificate Authority
CV	Card Verifiable
CVCA	Country Verifying Certificate Authority
DDO	Discretionary Data Object
DG	Data Group
DO	Data Object
EAC	Extended Access Control
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
DH	Diffie-Hellman
DST	Digital Signature Template
DV	Document Verifier
ICS	Implementation Conformance Statement (see Annex A)
IS	Inspection System
LDS	Logical Data Structure
KAEG	Key Agreement ElGamal-type

MRTD	Machine Readable Travel Document
MRZ	Machine Readable Zone
MSE	Manage Security Environment
OID	Object Identifier
PACE	Password Authenticated Connection Establishment
PIN	Personal Identification Number
PS	Pseudonymous Signature
PSA	Pseudonymous Signature for Authentication
PSO	Perform Security Operation
PUK	PIN Unblocking Key
RFU	Reserved for Future Use
RSA	Rivest Shamir Adleman
TA	Terminal Authentication

1.2 Reference documentation

The following documentation serves as a reference for this specification:

[R1] ICAO Doc 9303, Seventh Edition, Part 10

[R2] TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents – Part 1: eMRTDs with BAC/PACEv2 and EACv1, Version 2.10, March 2012

[R3] RFC 2119, S. Bradner, "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

[R4] ISO/IEC 7816-4:2013. Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange

[R5] PKCS #3: Diffie-Hellman Key-Agreement Standard, Version 1.4, November 1993

[R6] TR-03111: Technical Guideline, Elliptic Curve Cryptography, Version 2.0, June 2012

[R7] TR-03105: Technical Guideline, Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Version 1.1, July 2007, referencing EAC version 1.1

[R8] TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token – Part 3: Common Specifications, Version 2.21, December 2016

[R9] BSI, AFNOR: TR-03105 Part 3.2. Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Tests for Security Implementation, Version 1.5, 2018

[R10] RFC 3852, Housley, Russel, Cryptographic message syntax (CMS), RFC3852, 2004

[R11] ANSI, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography, ANSI X9.42-2000, 1999

1.3 Terminology

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [R3].

MUST	This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an

	absolute requirement of the specification.
MUST NOT	This phrase, or the phrase „SHALL NOT“, means 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“, means 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 MUST 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 MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1.4 Test Coverage

The following figure shows the test coverage of the different test specifications.

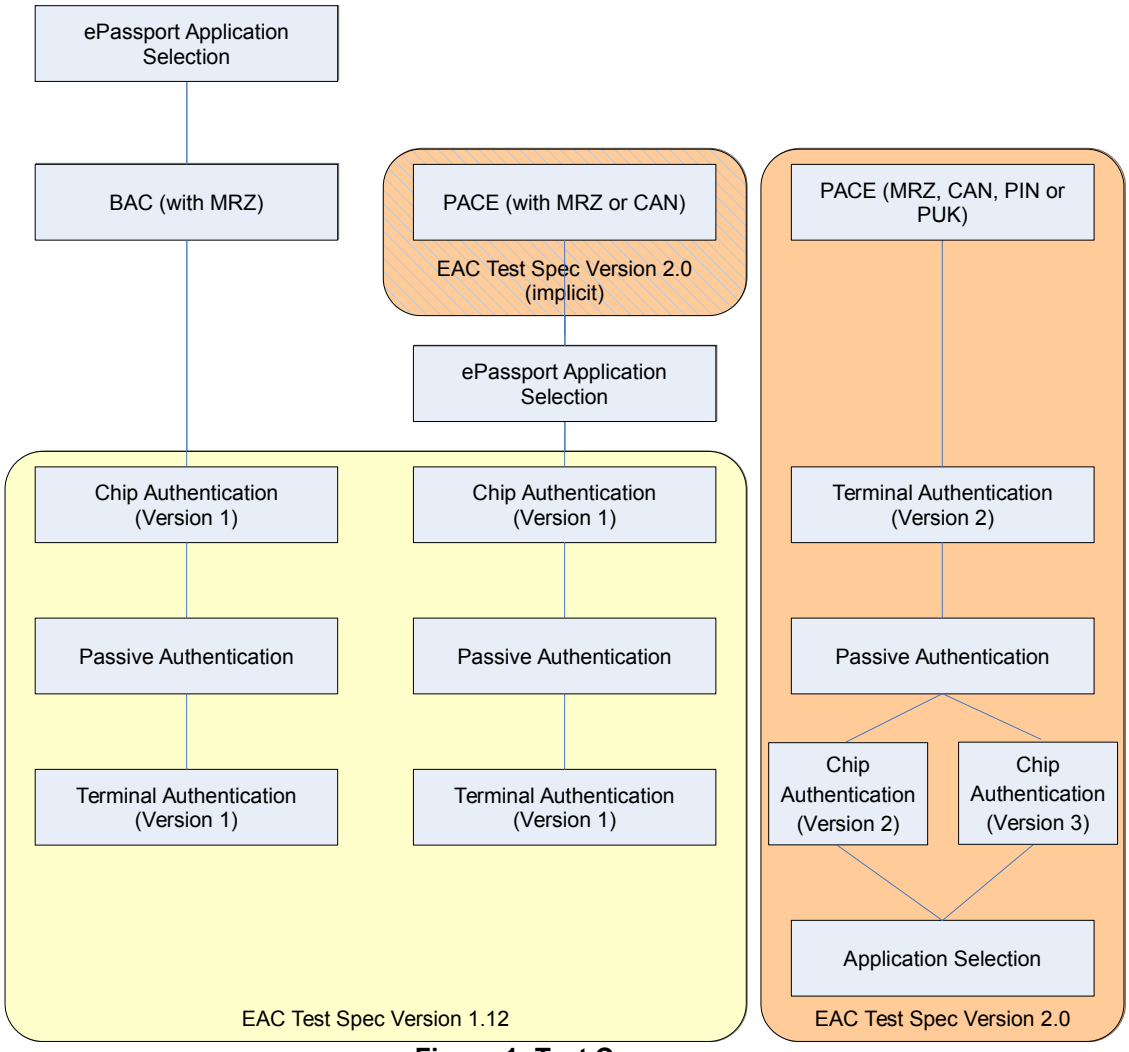

Figure 1: Test Coverage

The structure of the document is based on the EAC 1.11 test specification ([R9]). As far as possible identical unit names have been used for identical algorithm types, e.g. Chip Authentication is named as “Unit I” in this document as well as in the EAC 1.11 specification. The list of certificates is also based on the EAC 1.11 specification and extended by certificate types defined in the EAC 2.0 standard.

Three kinds of eID-Cards have to be observed as described below.

1.4.1 MRTD with BAC and EAC 1.x

If your MRTD is a BAC/EAC Version 1.x card please refer to [R9] only. There are no tests within this document that fit your needs.

1.4.2 MRTD with PACE and EAC 1.x

If your MRTD is an EAC Version 1.x card which also supports PACE, you have to perform the PACE tests defined here (see 3.3) and after that the following additional units of [R9]:

- ISO7816_H
- ISO7816_I
- ISO7816_J
- ISO7816_K
- ISO7816_L
- ISO7816_M

All test cases mentioned above have to be performed twice. In the first test run replace the precondition called “The BAC mechanism MUST/MUST NOT be performed” by “The PACE mechanism (with MRZ) MUST/MUST NOT be performed”. In the second test run replace the precondition called “The BAC mechanism MUST/MUST NOT be performed” by “The PACE mechanism (with CAN) MUST/MUST NOT be performed”.

Nevertheless PACE with EAC 1.x is not tested explicitly here.

1.4.3 eID-Card with EAC 2.x

If your MRTD is an eID-Card with EAC Version 2.x only, the test cases defined here have to be performed. If there are any references to other documents, they are described within the corresponding test unit.

1.4.4 ePassport Application Data Groups

The ePassport data groups are not tested within this specification.

1.4.5 eSign Application Data Groups

eSign is out of scope of both the EAC 1.x and EAC 2.x specification and therefore not tested here.

2 General test requirements

2.1 Test setup

For setting up these tests, any contactless reader supporting type A and type B protocols can be used. However, this reader has to support extended length APDUs requested for Terminal Authentication.

To execute any of the test cases described here, several types of test samples are required.

For executing all tests with one sample, this sample has to implement the ePassport application as defined by [R1], [R2] and [R8], the eID application as specified in [R9] and the eSign application for electronic signatures.

For executing separate tests of each application type one sample per application type is required (e.g. ePassport or eID card or eSign card). Cross-Application tests cannot be performed with these types of samples.

For executing cross-application mechanisms two types of samples are required: for executing read access tests to eID applications from ePassport application a sample with ePassport and eID application is necessary.

Some of the tests specified for layer 6 (ISO/IEC 7816) rely on the proper coding of the logical data structure stored in the chip. Therefore, it is RECOMMENDED that the layer 7 tests are performed before the layer 6 tests to detect coding related issues beforehand.

IMPORTANT NOTE: This test plan contains certain test cases, which verify the MRTD's behavior with expired certificates. During these tests, the effective date stored inside the chip is changed. Therefore a set of certificates can be used only once with a single card sample. After these tests have been performed, another sample or a new set of certificates is needed to repeat the tests.

This test plan also defines tests, which block or suspend PINs. After these tests have been performed, some of the features of the MRTD may be temporarily or permanently blocked or unusable.

Therefore, it is recommended to perform these tests as the last ones in a test sequence. If there is no way to unblock blocked or suspended PINs using PUKs or similar mechanisms, the vendor has to decide whether to perform or to skip these destructive tests.

2.2 Test profiles

This amendment defines several types of profiles. It is distinguished between “Application Profiles”, “Protocol Profiles”, “Algorithm Profiles” and “Data Group Profiles”. These types of profiles can be combined as defined by the corresponding card/application specification. Especially application profiles may include some implicit assumptions as defined in the corresponding specification (e.g. existence of PIN mechanisms when using eID application).

Profiles not mentioned within a test case MAY be present nevertheless (e.g. an eID application within ePassport tests). If the absence of a profile is necessary to fulfill the test case, it is separately mentioned in the test requirements.

2.2.1 Application Profiles

Profile-ID	Profile	Remark
ePassport	Electronic Passport Application	An application which contains data as specified in [R8] and [R1]. This profile implicitly includes usage of PACE with MRZ
eID	Electronic Identification Application	An application which contains data as specified in [R8]. This profile implicitly includes PIN/PUK management as

		defined in [R8].
eSign	Electronic Signature Application	An application which contains eSign specific data.

2.2.2 Protocol Profiles

Profile-ID	Profile	Remark
PACE	Password Authenticated Connection Establishment	A MRTD which does not contain sensitive biometric data, like finger prints, can still use the Password Authenticated Connection Establishment mechanism to support strong communication encryption. This profile only covers version 2.
TA2	Terminal Authentication, Version 2	Terminal Authentication MUST be performed for all EAC version 2 capable MRTDs within the general authentication procedure. This profile only covers version 2 Terminal Authentication.
CA2	Chip Authentication, Version 2	In addition to Terminal Authentication Chip authentication MUST be performed for all EAC version 2 capable MRTDs within the general authentication procedure. It supports chip cloning protection and strong communication encryption. This profile only covers version 2 Chip Authentication.
CA3	Chip Authentication, Version 3	This protocol is an alternative to the combination of Chip Authentication Version 2 and Restricted Identification (RI) providing additional features.
CA3_ReUse	Chip Authentication, Version 3 with Re-Use of PACE key	The ephemeral PACE-GM public key is reused by the chip during CA3.
MIG	Migration	According to the EAC specification the algorithm used for the Terminal Authentication process can be changed with an appropriate link certificate if the chip supports more than one algorithm. The tests for this Migration profile MUST only be performed, if the chip supports the migration from one cryptosystem to another. This must be stated in the ICS.
DATE	Date validation	Since the validation of the certificates effective and expiration date is not explicitly required by the EAC specification, the optional tests which belong to the Date validation profile must only be performed if this is supported by the chip. This must be stated in ICS.
RI	Restricted Identification	A MRTD which supports the Restricted Identification of terminals as specified in [R8].
RI_DP	Restricted Identification Domain Parameters	As RI. The MRTD additionally provides an optional RestrictedIdentificationDomainParameterInfo data structure. According to EAC specification, this is optional and must be stated in ICS.
AUX	Auxiliary Data Verification	A MRTD which supports Auxiliary Data Verification mechanisms (age verification, document validity

		verification or Municipality ID verification) as specified in [R8].
(NOT) CNG_PIN_PUK	Change PIN using PACE with PUK	This profile allows a “Change PIN“ procedure after PACE has been performed using PUK as authentication secret. Vice versa, if “Change PIN” procedure is NOT allowed, that profile is prefixed with NOT. According to EAC specification, this is optional and must be stated in ICS.
(NOT) CNG_PIN_AR	Change PIN allowed by Access Rights	This profile allows a “Change PIN” procedure for authentication terminals with “PIN Management” access right. Vice versa, if “Change PIN” procedure is NOT allowed, that profile is prefixed with NOT. According to EAC specification, this is optional and must be stated in ICS.
(NOT) CNG_CAN_AR	Change CAN allowed by Access Rights	This profile allows a “Change CAN” procedure for authentication terminals with “PIN Management” access right. Vice versa, if “Change CAN “procedure is NOT allowed, that profile is prefixed with NOT. According to EAC specification, this is optional and must be stated in ICS.
CS	Chip Security	A MRTD which stores a ChipSecurity file containing PrivilegedTerminalInfo with chip-individual keys and eIDSecurityInfo.
ENV	Envelope Mechanism	To support also terminals without extended length transport capability, chips can use the alternative Envelope mechanism.
CMP	Compare	The command Compare is used to verify authenticated auxiliary data.
AUTH_EXT	Authorization Extension	Authorization Extensions are special type of certificate extension. These extensions convey authorizations additional to those in the CHAT contained in the certificate.
BAC	Basic Access Control	According to ICAO Doc 9303 Part 11 starting on January 1st 2018 eMRTDs may support PACE only.
CSTA	EF.CardSecurity is protected by TA	EF.CardSecurity can be protected by PACE or by TA. Protection by TA was initially introduced in [R8] and protection by PACE was initially introduced in [R1].
PSAInfo	Use of PSAInfo	EF.CardAccess or EF.CardSecurity contains a PSAInfo element.
CardInfo	Use of CardInfo	EF.CardAccess or EF.CardSecurity contains a CardInfo element.
PrivTerInfo	Use of PrivilegedTerminalInfo	EF.CardAccess or EF.CardSecurity contains a PrivilegedTerminalInfo element.

2.2.3 Algorithm Profiles

Profile-ID	Profile	Remark
DH	Diffie-Hellman	According to the EAC specification, the chip can support Diffie-Hellman or elliptic curve based Diffie-Hellman key agreement algorithms. Test cases which belong to the DH profile are only applicable if the DH algorithm is used.
ECDH	Elliptic Curve Diffie-Hellman	According to the EAC specification, the chip can support Diffie-Hellman or elliptic curve based Diffie-Hellman key agreement algorithms. Test cases which belong to the ECDH profile are only applicable if the elliptic curve based DH algorithm is used.
ECDSA	Elliptic curve algorithm	According to the EAC specification a chip is free to support either elliptic curve or RSA based keys. All tests which belong to the ECDSA profile MUST only be processed if the test object is personalized with elliptic curve based keys.
RSA	RSA algorithm	According to the EAC specification a chip is free to support either elliptic curve or RSA based keys. All tests which belong to the RSA profile MUST only be processed if the test object is personalized with RSA based keys.

2.2.4 Data Group Profiles

If there are any (optional) data groups that have to be present to perform the corresponding tests, theses data groups are mentioned separately.

Profile-ID	Profile	Remark
DGx	Data Group x	Data group x must be present on the card

2.3 Key pair definition

The certificate sets defined in chapter 2.4 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 keys.

The initial CVCA root private keys SHOULD be provided by the ePassport vendor. It is also possible the ePassport vendor generates all keys and certificates on its own and passes it to the test operator for the tests.

There are separate CVCA roots for each terminal type. These CVCA roots have different key pairs.

For the key set 13 (CVCA_KEY_13, DV_KEY_13, IS_KEY_13) the algorithm for the cryptosystem migration MUST be used as defined in the ICS.

All key pairs MUST be generated independently, so it is not permitted to use the same key pair for all sets.

Key pair	
CVCA_KEY_00	The key pair CV_KEY_00 is the public/private key for the initial CVCA root.
DV_KEY_01	Key pair of the test DV 01
IS_KEY_01	Key pair of the test IS 01
DV_KEY_02	Key pair of the test DV 02

IS_KEY_02	Key pair of the test IS 02
DV_KEY_03	Key pair of the test DV 03
IS_KEY_03	Key pair of the test IS 03
DV_KEY_04	Key pair of the test DV 04
IS_KEY_04	Key pair of the test IS 04
DV_KEY_05	Key pair of the test DV 05
IS_KEY_05	Key pair of the test IS 05
DV_KEY_06	Key pair of the test DV 06
IS_KEY_06	Key pair of the test IS 06
CVCA_KEY_07	Key pair of the test CVCA 07
DV_KEY_07	Key pair of the test DV 07
IS_KEY_07	Key pair of the test IS 07
CVCA_KEY_08	Key pair of the test CVCA 08
CVCA_KEY_09	Key pair of the test CVCA 09
DV_KEY_09	Key pair of the test DV 09
CVCA_KEY_10	Key pair of the test CVCA 10
DV_KEY_10	Key pair of the test DV 10
IS_KEY_10	Key pair of the test IS 10
CVCA_KEY_11	Key pair of the test CVCA 11
DV_KEY_11	Key pair of the test DV 11
IS_KEY_11	Key pair of the test IS 11
DV_KEY_12	Key pair of the test DV 12
CVCA_KEY_13	Key pair of the test CVCA 13
DV_KEY_13	Key pair of the test DV 13
IS_KEY_13	Key pair of the test IS 13
DV_KEY_14a	Key pair of the test DV 14 (length equal to CVCA Key length)
DV_KEY_14b	Key pair of the test DV 14 (MUST be shorter than CVCA Key length)
IS_KEY_14a	Key pair of the test IS 14 (length equal to CVCA Key length)
IS_KEY_14b	Key pair of the test IS 14 (MUST be shorter than CVCA Key length)
DV_KEY_15	Key pair of the test DV 15
IS_KEY_15	Key pair of the test IS 15
DV_KEY_16	Key pair of the test DV 16
IS_KEY_16	Key pair of the test IS 16
CVCA_KEY_17	The key pair CVCA_KEY_17 is the public/private key for the AT CVCA root
DV_KEY_17	Key pair of the test DV 17
AT_KEY_17	Key pair of the test AT 17
DV_KEY_18	Key pair of the test DV 18
AT_KEY_18	Key pair of the test AT 18
DV_KEY_19	Key pair of the test DV 19
AT_KEY_19	Key pair of the test AT 19

DV_KEY_20	Key pair of the test DV 20
AT_KEY_20	Key pair of the test AT 20
DV_KEY_21	Key pair of the test DV 21
AT_KEY_21	Key pair of the test AT 21
DV_KEY_22	Key pair of the test DV 22
AT_KEY_22	Key pair of the test AT 22
AT_CVCA_KEY_23a	Key pair of the test AT CVCA 23a
AT_CVCA_KEY_23b	Key pair of the test AT CVCA 23b
DV_KEY_23	Key pair of the test DV 23
DV_KEY_24	Key pair of the test DV 24
AT_KEY_24	Key pair of the test AT 24
DV_KEY_25	deleted in version 1.00 RC
IS_KEY_25	deleted in version 1.00 RC
DV_KEY_26	deleted in version 1.00 RC
IS_KEY_26	deleted in version 1.00 RC
DV_KEY_27	deleted in version 1.1
IS_KEY_27	deleted in version 1.1
DV_KEY_28	deleted in version 1.1
IS_KEY_28	deleted in version 1.1
DV_KEY_30	Key pair of the test DV_30
AT_KEY_30	Key pair of the test AT_30
DV_KEY_31	Key pair of the test DV_31
AT_KEY_31	Key pair of the test AT_31

2.4 Certificate specification

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 passport samples, which do already have a preconfigured trust point based on their own CVCA key pair, the chapters below defines a set of certificates relative to the effective date ($CVCA_{eff}$) and expiration date($CVCA_{exp}$) of the given CVCA. The time span between $CVCA_{eff}$ and $CVCA_{exp}$ MUST be at least two month to allow proper adoption of the certificate time scheme defined below. The “current date” of the provided sample MUST be set to $CVCA_{eff}$ before the tests are started. The CVCA MUST NOT restrict authorization in any way, i.e. its Certificate Holder Authorization contains all rights. The provider of the sample or the test laboratory has to generate the corresponding certificate according to this specification based on the CVCA data.

There are separate CVCA roots for each terminal type, but they all SHOULD have equal effective and expiration dates.

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.

2.4.1 Certificate Set 1

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

2.4.1.1 DV_CERT_1

ID	DV_CERT_1
Purpose	This certificate is a regular DV certificate, which validity period starts at the effective date of the CVCA and expires after one month.
Version	1.11
Referred by	Test case EAC2_ISO7816_I_1, Test case EAC2_ISO7816_I_2, Test case EAC2_ISO7816_I_3, Test case EAC2_ISO7816_I_4, Test case EAC2_ISO7816_I_5, Test case EAC2_ISO7816_I_6, Test case EAC2_ISO7816_I_7, Test case EAC2_ISO7816_I_8, Test case EAC2_ISO7816_I_9, Test case EAC2_ISO7816_I_10, Test case EAC2_ISO7816_I_11, Test case EAC2_ISO7816_I_12, Test case EAC2_ISO7816_I_13, Test case EAC2_ISO7816_I_14, Test case EAC2_ISO7816_I_15, Test case EAC2_ISO7816_I_16, Test case EAC2_ISO7816_J_1, Test case EAC2_ISO7816_J_2, Test case EAC2_ISO7816_J_3, Test case EAC2_ISO7816_J_4, Test case EAC2_ISO7816_J_5, Test case EAC2_ISO7816_J_12, Test case EAC2_ISO7816_J_14, Test case EAC2_ISO7816_J_15, Test case EAC2_ISO7816_J_16, Test case EAC2_ISO7816_J_20, Test case EAC2_ISO7816_J_23, Test case EAC2_ISO7816_J_24, Test case EAC2_ISO7816_J_25, Test case EAC2_ISO7816_J_26, Test case EAC2_ISO7816_J_27, Test case EAC2_ISO7816_J_28, Test case EAC2_ISO7816_J_29, Test case EAC2_ISO7816_J_30, Test case EAC2_ISO7816_J_31, Test case EAC2_ISO7816_J_32, Test case EAC2_ISO7816_J_33, Test case EAC2_ISO7816_J_34, Test case EAC2_ISO7816_J_35, Test case EAC2_ISO7816_J_36, Test case EAC2_ISO7816_J_37, Test case EAC2_ISO7816_J_38, Test case EAC2_ISO7816_J_39, Test case EAC2_ISO7816_K_1, Test case EAC2_ISO7816_K_2, Test case EAC2_ISO7816_K_3, Test case EAC2_ISO7816_K_4, Test case EAC2_ISO7816_K_6, Test case EAC2_ISO7816_K_7, Test case EAC2_ISO7816_K_8, Test case EAC2_ISO7816_K_9, Test case EAC2_ISO7816_K_10, Test case EAC2_ISO7816_K_11, Test case EAC2_ISO7816_K_12, Test case EAC2_ISO7816_K_14, Test case EAC2_ISO7816_L_9, Test case EAC2_ISO7816_L_10, Test case EAC2_ISO7816_L_11, Test case EAC2_ISO7816_L_12 The DV_CERT_1 SHOULD also be used for all other test cases that rely on an established EAC session to access DG3 and DG4 of ePassports.
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd*

	7F 49 *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certification Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	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_01
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.1.2 DV_CERT_1a

ID	DV_CERT_1a
Purpose	This certificate is similar to DV_CERT_1, but does not contain a Certificate Holder Authorization
Version	1.11
Referred by	Test case EAC2_ISO7816_J_6
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	absent
	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

2.4.1.3 DV_CERT_1b

ID	DV_CERT_1b
Purpose	This certificate is similar to DV_CERT_1, but does not contain a Certificate Effective Date
Version	1.11
Referred by	Test case EAC2_ISO7816_J_7
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx bytes)

	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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	absent
	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

2.4.1.4 DV_CERT_1c

ID	DV_CERT_1c	
Purpose	This certificate is similar to DV_CERT_1, but does not contain a Certificate Expiration Date	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_8	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **5F 25** 06 *gg* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx bytes) *gg* is the placeholder for the BCD encoded effective 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$

	Certificate expiration date	absent
	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

2.4.1.5 DV_CERT_1d

ID	DV_CERT_1d	
Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Effective Date (Invalid BCD encoding)	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_9	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **5F 25** 06 0A 0B 0C 0D 0E 0F **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx bytes) *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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	0A 0B 0C 0D 0E 0F (invalid BCD encoding)
	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

2.4.1.6 DV_CERT_1e

ID	DV_CERT_1e

Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Expiration Date(Invalid BCD encoding)	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_10	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **5F 25** 06 *gg* **5F 24** 06 0A 0B 0C 0D 0E 0F **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx bytes) *gg* is the placeholder for the BCD encoded effective 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	0A 0B 0C 0D 0E 0F (invalid BCD encoding)
	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

2.4.1.7 DV_CERT_1f

ID	DV_CERT_1f
Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Effective Date (Invalid Gregorian date)
Version	1.11
Referred by	Test case EAC2_ISO7816_J_17
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00

	42 *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	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	$CVCA_{exp}$
	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

2.4.1.8 DV_CERT_1g

ID	DV_CERT_1g
Purpose	This certificate is similar to DV_CERT_1, but contains a badly encoded Certificate Expiration Date (Invalid Gregorian date)
Version	1.11
Referred by	Test case EAC2_ISO7816_J_18
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83

	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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	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	Public key of key pair DV_KEY_01
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.1.9 DV_CERT_1h

ID	DV_CERT_1h
Purpose	This certificate is similar to DV_CERT_1, but contains a Certificate Expiration Date BEFORE the Certificate Effective Date
Version	1.11
Referred by	Test case EAC2_ISO7816_J_19
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 day
	Certificate expiration date	$CVCA_{eff}$
	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

2.4.1.10 DV_CERT_1i

ID	DV_CERT_1i
Purpose	This certificate is similar to DV_CERT_1, but contains a Certificate Holder Authorization with an invalid combination of OID (<id-AT>) and discretionary data object (structured like a relative authorization bit map for an IS)
Version	1.11
Referred by	Test case EAC2_ISO7816_J_21
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes),

	xx is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	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_01
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.1.11 DV_CERT_1j

ID	DV_CERT_1j
Purpose	This certificate is similar to DV_CERT_1, but contains a Public Key with an invalid OID
Version	1.12
Referred by	Test case EAC2_ISO7816_J_22
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE001
	Certificate Public Key	Bad OID (Use 0.4.0.127.0.7.2.2.2.5.1)
	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_01
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.1.12 IS_CERT_1

ID	IS_CERT_1
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_1
Version	1.11
Referred by	Test case EAC2_ISO7816_I_1, Test case EAC2_ISO7816_I_2, Test case EAC2_ISO7816_I_3, Test case EAC2_ISO7816_I_4, Test case EAC2_ISO7816_I_5, Test case EAC2_ISO7816_I_6, Test case EAC2_ISO7816_I_7, Test case EAC2_ISO7816_I_8, Test case EAC2_ISO7816_I_9, Test case EAC2_ISO7816_I_10, Test case EAC2_ISO7816_I_12, Test case EAC2_ISO7816_I_13, Test case EAC2_ISO7816_I_14, Test case EAC2_ISO7816_I_15, Test case EAC2_ISO7816_I_16, Test case EAC2_ISO7816_J_1, Test case EAC2_ISO7816_J_2, Test case EAC2_ISO7816_J_3, Test case EAC2_ISO7816_J_4, Test case EAC2_ISO7816_J_5, Test case EAC2_ISO7816_J_6, Test case EAC2_ISO7816_J_7, Test case EAC2_ISO7816_J_8, Test case EAC2_ISO7816_J_9, Test case EAC2_ISO7816_J_10, Test case EAC2_ISO7816_J_15, Test case EAC2_ISO7816_J_16, Test case EAC2_ISO7816_J_17, Test case EAC2_ISO7816_J_18, Test case EAC2_ISO7816_J_19, Test case EAC2_ISO7816_J_20, Test case EAC2_ISO7816_J_21, Test case EAC2_ISO7816_J_22, Test case EAC2_ISO7816_K_1, Test case EAC2_ISO7816_K_2, Test case EAC2_ISO7816_K_3, Test case EAC2_ISO7816_K_6, Test case EAC2_ISO7816_K_7, Test case EAC2_ISO7816_K_8, Test case EAC2_ISO7816_K_9, Test case EAC2_ISO7816_K_10, Test case EAC2_ISO7816_K_11, Test case EAC2_ISO7816_K_12, Test case EAC2_ISO7816_K_14, Test case EAC2_ISO7816_L_9, Test case EAC2_ISO7816_L_10, Test case EAC2_ISO7816_L_11, Test case EAC2_ISO7816_L_12
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE001
	Certificate Holder Reference	DETESTISDE001
	Certificate Holder Authorization	IS, DG 3, DG 4
	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

2.4.2 Certificate Set 2

This certificate set contains certificates which are used to verify the behavior of ePassports in respect to foreign IS certificates.

2.4.2.1 DV_CERT_2

ID	DV_CERT_2
Purpose	This certificate is a regular foreign DV certificate, which validity period starts at the effective date of the CVCA and expires after one month.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_11
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 43 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE002
	Certificate Holder Authorization	foreign 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

2.4.2.2 IS_CERT_2a

ID	IS_CERT_2a
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_2. It has an advanced effective date. (Beyond the expiration date of IS_CERT_2b).
Version	1.11
Referred by	Test case EAC2_ISO7816_J_11
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes),

	xx is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE002
	Certificate Holder Reference	DETESTISDE002
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 14 days
	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

2.4.2.3 IS_CERT_2b

ID	IS_CERT_2b
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_2. It has an expiration date BEFORE the effective date of IS_CERT_2a.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_11
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE002
	Certificate Holder Reference	DETESTISDE002
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	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

2.4.3 Certificate Set 3

The certificate set follows a certification scheme where the DV permits full access to data group 3 and 4 while the IS certificate restricts the access to specific data group.

2.4.3.1 DV_CERT_3

ID	DV_CERT_3
Purpose	This certificate is a regular DV certificate, with access rights for both data group 3 and 4.
Version	1.11
Referred by	Test case EAC2_ISO7816_L_1, Test case EAC2_ISO7816_L_2, Test case EAC2_ISO7816_L_3, Test case EAC2_ISO7816_L_4, Test case EAC2_ISO7816_O_1, Test case EAC2_ISO7816_O_2
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE003
	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_03
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.3.2 DV_CERT_3a

ID	DV_CERT_3a	
Purpose	This certificate is a regular DV certificate, with access rights for both data group 3 and 4. It is a copy of DV_CERT_3 with the exception that all RFU bits within CHAT are set to 1.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_L_35	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 BF **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE003
	Certificate Holder Authorization	domestic DV, DG 3, DG 4, RFU=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

2.4.3.3 IS_CERT_3a

ID	IS_CERT_3a	
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_3. It encodes access rights for data group 3 only.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_K_13, Test case EAC2_ISO7816_K_14, Test case EAC2_ISO7816_K_15, Test case EAC2_ISO7816_L_1, Test case EAC2_ISO7816_L_2, Test case EAC2_ISO7816_M_6, Test case EAC2_ISO7816_O_1	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 01 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE003
	Certificate Holder Reference	DETESTISDE003
	Certificate Holder Authorization	IS, DG 3
	Certificate effective date	$CVCA_{eff}$
	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

2.4.3.4 IS_CERT _3b

ID	IS_CERT_3b	
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_3. It encodes access rights for data group 4 only.	
Version	1.11	
Referred by		
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 02 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE003
	Certificate Holder Reference	DETESTISDE003
	Certificate Holder Authorization	IS, DG 4
	Certificate effective date	$CVCA_{eff}$
	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

2.4.3.5 IS_CERT_3c

ID	IS_CERT_3c
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_3. It encodes access rights for data group 3 only. It is a copy of IS_CERT_3a with the exception that all RFU bits within CHAT are set to 1.
Version	EAC2_1.0

Referred by	Test case EAC2_ISO7816_L_35	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 3D **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE003
	Certificate Holder Reference	DETESTISDE003
	Certificate Holder Authorization	IS, DG 3, RFU=1
	Certificate effective date	$CVCA_{eff}$
	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

2.4.4 Certificate Set 4

The certificate set follows a certification scheme where the DV permits only access to data group 3 while the IS certificate permits full access to data group 3 and 4.

2.4.4.1 DV_CERT_4

ID	DV_CERT_4
Purpose	This certificate is a regular DV certificate, with access rights for group 3 only.
Version	1.11
Referred by	Test case EAC2_ISO7816_L_5, Test case EAC2_ISO7816_L_6, Test case EAC2_ISO7816_O_3

<table>
<tr><td>Content definition</td><td colspan="2">7F 21 aa
  7F 4E bb
    5F 29 01 00
    42 cc dd
    7F 49 ee ff
    5F 20 xx yy
    7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 81
    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 of the certificate body object
cc is the encoded length of the Certification Authority Reference
dd is the placeholder for the Certification Authority Reference (cc bytes)
ee is the encoded length of the certificate's public key,
ff is the placeholder for the certificate's public key bytes (ee bytes),
xx is the encoded length of the Certificate Holder Reference
yy is the placeholder for the Certificate Holder Reference (xx 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)</td></tr>
<tr><td rowspan="7">Parameter</td><td>Certification Authority Reference</td><td>As defined by the CVCA</td></tr>
<tr><td>Certificate Holder Reference</td><td>DETESTDVDE004</td></tr>
<tr><td>Certificate Holder Authorization</td><td>domestic DV, DG 3</td></tr>
<tr><td>Certificate effective date</td><td>$CVCA_{eff}$</td></tr>
<tr><td>Certificate expiration date</td><td>$CVCA_{eff}$ + 1 month</td></tr>
<tr><td>Public Key reference</td><td>Public key of key pair DV_KEY_04</td></tr>
<tr><td>Signing Key reference</td><td>Signed with the private key of key pair CVCA_KEY_00</td></tr>
</table>

2.4.4.2 IS_CERT_4

<table>
<tr><td>ID</td><td>IS_CERT_4</td></tr>
<tr><td>Purpose</td><td>This certificate is a regular IS certificate, which is issued by the DV_CERT_4. It encodes access rights for data group 3 and data group 4.</td></tr>
<tr><td>Version</td><td>1.11</td></tr>
<tr><td>Referred by</td><td>Test case EAC2_ISO7816_L_5, Test case EAC2_ISO7816_L_6, Test case EAC2_ISO7816_O_3</td></tr>
<tr><td>Content definition</td><td>7F 21 aa
  7F 4E bb
    5F 29 01 00
    42 cc dd
    7F 49 ee ff
    5F 20 xx yy</td></tr>
</table>

	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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE004
	Certificate Holder Reference	DETESTISDE004
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair IS_KEY_04
	Signing Key reference	Signed with the private key of key pair DV_KEY_04

2.4.5 Certificate Set 5

The certificate set follows a certification scheme where the DV permits only access to data group 4 while the IS certificate permits full access to data group 3 and 4.

2.4.5.1 DV_CERT_5

ID	DV_CERT_5
Purpose	This certificate is a regular DV certificate, with access rights for group 4 only.
Version	1.11
Referred by	Test case EAC2_ISO7816_L_7, Test case EAC2_ISO7816_L_8, Test case EAC2_ISO7816_O_4
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 82

	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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE005
	Certificate Holder Authorization	domestic DV, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_05
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.5.2 IS_CERT_5

ID	IS_CERT_5
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_5. It encodes access rights for data group 3 and data group 4.
Version	1.11
Referred by	Test case EAC2_ISO7816_L_7, Test case EAC2_ISO7816_L_8, Test case EAC2_ISO7816_O_4
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object

	cc is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE005
	Certificate Holder Reference	DETESTISDE005
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	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

2.4.6 Certificate Set 6

This certificate set contains certificate which have different effective and expiration dates to test the ePassports behavior in respect to the update of the effective date and with expired certificates.

2.4.6.1 DV_CERT_6

ID	DV_CERT_6
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	1.11
Referred by	Test case EAC2_ISO7816_M_1, Test case EAC2_ISO7816_M_2
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes)

	ee is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE006
	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_06
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.6.2 DV_CERT_6a

ID	DV_CERT_6a
Purpose	This DV certificate is similar to DV_CERT_6, but the certificate effective date is beyond the DV_CERT_6 expiration date.
Version	1.11
Referred by	Test case EAC2_ISO7816_M_2
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE006
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 1 day
	Certificate expiration date	$CVCA_{eff}$ + 2 month
	Public Key reference	Public key of key pair DV_KEY_06
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.6.3 IS_CERT_6a

ID	IS_CERT_6a	
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_6. This IS certificate has an advanced effective date. (Beyond the expiration date of IS_CERT_6b)	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_1	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE006
	Certificate Holder Reference	DETESTISDE006
	Certificate Holder Authorization	IS, DG 3, DG 4

	Certificate effective date	$CVCA_{eff}$ + 14 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair IS_KEY_06
	Signing Key reference	Signed with the private key of key pair DV_KEY_06

2.4.6.4 IS_CERT_6b

ID	IS_CERT_6b	
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_6. This IS certificate has an expiration date BEFORE the effective date of IS_CERT_6a.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_1	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE006
	Certificate Holder Reference	DETESTISDE006
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	Public Key reference	Public key of key pair IS_KEY_06
	Signing Key reference	Signed with the private key of key pair DV_KEY_06

2.4.7 Certificate Set 7

This certificate set defines a link certificate used for the tests about the trust point update mechanism.

2.4.7.1 LINK_CERT_7

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

ID	LINK_CERT_7	
Purpose	This certificate is a link certificate, which validity period starts one day before the original CVCA certificate expires.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_3	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *optional: 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTLINKDE007
	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_07

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00
	Certificate Extension	As defined by the CVCA

2.4.7.2 DV_CERT_7a

ID	DV_CERT_7a	
Purpose	This certificate is a domestic DV certificate, which was issued by the original CVCA.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_3	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the original CVCA
	Certificate Holder Reference	DETESTDVDE007
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{exp}$
	Public Key reference	Public key of key pair DV_KEY_07
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.7.3 DV_CERT_7b

ID	DV_CERT_7b

Purpose	This certificate is a domestic DV certificate, which was issued by the update CVCA (LINK_CERT_7).	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_3, Test case EAC2_ISO7816_M_8	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTLINKDE007
	Certificate Holder Reference	DETESTDVDE007
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 1 day
	Certificate expiration date	$CVCA_{exp}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_07
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_07

2.4.8 Certificate Set 8

This certificate set defines a link certificate used for the tests about the trust point update mechanism.

Note for ECDSA profile: Since the cryptographic mechanism is not changed by the link certificates defined in this certificate set, it must be stated by the vendor of the test sample if the domain parameters should be included. (see ICS A).

2.4.8.1 LINK_CERT_8

This link certificate is used to update the trust point defined by LINK_CERT_7.

ID	LINK_CERT_8	
Purpose	This certificate is a link certificate, based on the LINK_CERT_7	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_4	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *optional: 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETESTLINKDE007
	Certificate Holder Reference	DETESTLINKDE008
	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_08
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_07
	Certificate Extension	As defined by CVCA

2.4.9 Certificate Set 9

2.4.9.1 LINK_CERT_9

ID	LINK_CERT_9	
Purpose	This certificate is a link certificate, based on the LINK_CERT_8	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_4	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *optional: 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETESTLINKDE008
	Certificate Holder Reference	DETEST_LINKDE009
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 3 month
	Certificate expiration date	$CVCA_{exp}$ + 6 month
	Public Key reference	Public key of key pair CVCA_KEY_09
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_08
	Certificate Extension	As defined by CVCA

2.4.9.2 DV_CERT_9

ID	DV_CERT_9	
Purpose	This certificate is a domestic DV certificate, which was issued by LINK_CERT_9.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_4	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETEST_LINKDE009
	Certificate Holder Reference	DETESTDVDE009
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 3 month
	Certificate expiration date	$CVCA_{exp}$ + 4 month
	Public Key reference	Public key of key pair DV_KEY_09
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_09

2.4.10 Certificate Set 10

2.4.10.1 LINK_CERT_10

ID	LINK_CERT_10
Purpose	This certificate is an irregular IS CVCA certificate. The signing key is a DV key.
Version	1.11

Referred by	Test case EAC2_ISO7816_J_41, Test case EAC2_ISO7816_J_42	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *optional: 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETESTDVDE010
	Certificate Holder Reference	As defined by the initial 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_10
	Certificate Extension	As defined by CVCA

2.4.10.2 DV_CERT_10a

ID	DV_CERT_10a
Purpose	This certificate is a regular domestic DV certificate, which validity period starts at the effective date of the CVCA and expires after one month.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_41, Test case EAC2_ISO7816_J_43, Test case EAC2_ISO7816_J_44

Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE010
	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_10
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.10.3 DV_CERT_10b

ID	DV_CERT_10b
Purpose	This certificate is a regular foreign DV certificate, which validity period starts at the effective date of the CVCA and expires after one month.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_42, Test case EAC2_ISO7816_J_45, Test case EAC2_ISO7816_J_46
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy*

	7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 43 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE010
	Certificate Holder Authorization	foreign 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_10
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.10.4 DV_CERT_10c

ID	DV_CERT_10c
Purpose	This certificate is an irregular DV domestic certificate. The signing key is a DV key.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_43, Test case EAC2_ISO7816_J_45
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object

	cc is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE010
	Certificate Holder Reference	DETESTDVDE110
	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_10
	Signing Key reference	Signed with the private key of key pair DV_KEY_10

2.4.10.5 DV_CERT_10d

ID	DV_CERT_10d
Purpose	This certificate is an irregular DV foreign certificate. The signing key is a DV key.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_44, Test case EAC2_ISO7816_J_46
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 43 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE010
	Certificate Holder Reference	DETESTDVDE110
	Certificate Holder Authorization	foreign 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_10
	Signing Key reference	Signed with the private key of key pair DV_KEY_10

2.4.10.6 IS_CERT_10

ID	IS_CERT_10	
Purpose	This certificate is an irregular domestic IS certificate. This IS certificate is signed by the CVCA key.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_40	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTISDE010
	Certificate Holder Authorization	IS, DG 3, DG 4

	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	Public Key reference	Public key of key pair IS_KEY_10
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.11 Certificate Set 11

2.4.11.1 LINK_CERT_11a

ID	LINK_CERT_11a	
Purpose	This certificate is an irregular IS CVCA certificate. The signing key is an IS key.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_50	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETESTISDE011
	Certificate Holder Reference	As defined by the initial 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_11
	Certificate Extension	As defined by CVCA

2.4.11.2 LINK_CERT_11b

ID	LINK_CERT_11b	
Purpose	This certificate is a valid link certificate.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_5	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *optional: 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETEST_LINKDE009
	Certificate Holder Reference	DETEST_LINKDE011
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 5 months
	Certificate expiration date	$CVCA_{exp}$ + 8 months
	Public Key reference	Public key of key pair CVCA_KEY_11

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_09
	Certificate Extension	As defined by CVCA

2.4.11.3 DV_CERT_11a

ID	DV_CERT_11a	
Purpose	This certificate is a regular domestic DV certificate, which validity period starts at the effective date of the CVCA and expires after one month.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_47, Test case EAC2_ISO7816_J_48, Test case EAC2_ISO7816_J_49, Test case EAC2_ISO7816_J_50	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE011
	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_11
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.11.4 DV_CERT_11b

ID	DV_CERT_11b	
Purpose	This certificate is an irregular foreign DV certificate. The signing key is an IS key.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_47	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 43 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTISDE011
	Certificate Holder Reference	DETESTDVDE011
	Certificate Holder Authorization	foreign 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_11
	Signing Key reference	Signed with the private key of key pair IS_KEY_11

2.4.11.5 DV_CERT_11c

ID	DV_CERT_11c
Purpose	This certificate is an irregular domestic DV certificate. The signing key is an IS key.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_48
Content definition	**7F 21** *aa* **7F 4E** *bb*

	5F 29 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTISDE011
	Certificate Holder Reference	DETESTDVDE011
	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_11
	Signing Key reference	Signed with the private key of key pair IS_KEY_11

2.4.11.6 DV_CERT_11d

ID	DV_CERT_11d
Purpose	This certificate is a regular domestic DV certificate, which validity period starts at the effective date of the referencing CVCA 11b and expires after one month.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_M_8
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETEST_LINKDE011
	Certificate Holder Reference	DETESTDVDE011
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 5 months
	Certificate expiration date	$CVCA_{exp}$ + 6 months
	Public Key reference	Public key of key pair DV_KEY_11
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_11

2.4.11.7 IS_CERT_11a

ID	IS_CERT_11a
Purpose	This certificate is a regular IS certificate.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_47, Test case EAC2_ISO7816_J_48, Test case EAC2_ISO7816_J_49, Test case EAC2_ISO7816_J_50
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes)

	ee is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE011
	Certificate Holder Reference	DETESTISDE011
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	Public Key reference	Public key of key pair IS_KEY_11
	Signing Key reference	Signed with the private key of key pair DV_KEY_11

2.4.11.8 IS_CERT_11b

ID	IS_CERT_11b
Purpose	This certificate is an irregular IS certificate. The signing key is an IS key.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_49
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTISDE011
	Certificate Holder Reference	DETESTISDE111
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 13 days
	Public Key reference	Public key of key pair IS_KEY_11
	Signing Key reference	Signed with the private key of key pair IS_KEY_11

2.4.11.9 IS_CERT_11c

ID	IS_CERT_11c	
Purpose	This certificate is an irregular IS certificate. The signing key is a CVCA key.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_M_5	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETEST_LINKDE011
	Certificate Holder Reference	DETESTISDE011
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 5 months
	Certificate expiration date	$CVCA_{exp}$ + 6 months

	Public Key reference	Public key of key pair IS_KEY_11
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_11

2.4.12 Certificate Set 12

This certificate set is used for the certificate structure tests.

2.4.12.1 DV_CERT_12a

ID	DV_CERT_12a	
Purpose	This certificate is a domestic DV certificate.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_23, Test case EAC2_ISO7816_J_33	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.2 DV_CERT_12b

ID	DV_CERT_12b	
Purpose	Certificate with a wrong “certificate body” tag	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_24	
Content definition	**7F 21** *aa* **7F 4F** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.3 DV_CERT_12c

ID	DV_CERT_12c
Purpose	Certificate with a wrong “certificate signature” tag
Version	1.11
Referred by	Test case EAC2_ISO7816_J_25
Content definition	**7F 21** *aa* **7F 4E** *bb*

	5F 29 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **5F 25** 06 *gg* **5F 24** 06 *hh* **5F 38** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.4 DV_CERT_12d

ID	DV_CERT_12d
Purpose	Certificate with an inconsistent “certificate body” DO (wrong length)
Version	1.11
Referred by	Test case EAC2_ISO7816_J_26
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object **decreased by one** *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.5 DV_CERT_12e

ID	DV_CERT_12e
Purpose	Certificate with an inconsistent “certificate signature” DO (The length byte specifies one by less than the actual signature length)
Version	1.11
Referred by	Test case EAC2_ISO7816_J_27
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key,

	ff is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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 **decreased by one**, *jj* is the placeholder for the certificates signature (ii + 1 bytes)	
Parameter	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.6 DV_CERT_12f

ID	DV_CERT_12f
Purpose	Certificate with a wrong signature
Version	1.11
Referred by	Test case EAC2_ISO7816_J_28
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) **last byte is increased by one (mod 256)**

Parameter	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.7 DV_CERT_12g

ID	DV_CERT_12g	
Purpose	Certificate with a wrong signature	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_29	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) – **last byte is dropped and ii is updated according to the new length**	
Parameter	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days

	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.8 DV_CERT_12h

ID	DV_CERT_12h	
Purpose	Modification in the certificate public key: OID is missing	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_35	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes) – **it does not contain any OID DO**, *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.9 DV_CERT_12i

ID	DV_CERT_12i

Purpose	Modification in the certificate public key: wrong OID	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_34	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes) – **the OID has an incorrect value that does not indicate id-TA: (0.4.0.127.0.7.2.2.3.x.y)**, *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.10 DV_CERT_12j

ID	DV_CERT_12j
Purpose	**For ECDSA profile only:** Modification in the certificate public key: the elliptic curve public point is missing
Version	1.11
Referred by	Test case EAC2_ISO7816_J_36
Content definition	**7F 21** *aa* **7F 4E** *bb*

	5F 29 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes) – **The elliptic curve public point is missing,** *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.11 DV_CERT_12k

ID	DV_CERT_12k
Purpose	**For RSA profile only:** Modification in the certificate public key: the RSA modulus is missing
Version	1.11
Referred by	Test case EAC2_ISO7816_J_37
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83

	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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes) – **The RSA modulus is missing**, *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.12 DV_CERT_12l

ID	DV_CERT_12l
Purpose	**For RSA profile only:** Modification in the certificate public key: the RSA public exponent is missing
Version	1.11
Referred by	Test case EAC2_ISO7816_J_38
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes) – **The RSA public exponent is missing**, *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.13 DV_CERT_12m

ID	DV_CERT_12m
Purpose	Modification in the certificate public key: For ECDSA profile: an unknown DO is present within the EC parameters (tag '77'), For RSA profile: an unknown DO is present within the RSA parameters ('77 01 00')
Version	1.11
Referred by	Test case EAC2_ISO7816_J_39
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes) – **An unknown**

	DO '77' is present *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.14 DV_CERT_12n

ID	DV_CERT_12n
Version	Has been merged with DV_CERT_12m in version 1.1

2.4.12.15 DV_CERT_12o

ID	DV_CERT_12o
Purpose	**For RSA profile only:** Certificate with a wrong signature
Version	1.11
Referred by	Test case EAC2_ISO7816_J_30
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference

	yy is the placeholder for the Certificate Holder Reference (xx 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) – **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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.16 DV_CERT_12p

ID	DV_CERT_12p
Purpose	**For ECDSA profile only:** 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.
Version	1.11
Referred by	**Test case EAC2_ISO7816_J_31**
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) – **with r = 0**	
Parameter	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.12.17 DV_CERT_12q

ID	DV_CERT_12q	
Purpose	**For ECDSA profile only:** 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.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_32	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) – **with s = 0**	
Parameter	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE012
	Certificate Holder Authorization	domestic DV, DG 3, DG 4

	Certificate effective date	$CVCA_{eff}$ + 1 month + 20 days
	Certificate expiration date	$CVCA_{eff}$ + 1 month + 25 days
	Public Key reference	Public key of key pair DV_KEY_12
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.13 Certificate Set 13

This certificate set defines a link certificate used to update the chip signature mechanism according to the migration policy as defined by the manufacturer. The cryptographic elements of these certificates MUST use the new mechanisms besides the signature of the LINK_CERT_13 which is done with the original signature mechanism. This certificate set is only needed if the "Migration" profile is supported.

2.4.13.1 LINK_CERT_13

Note for ECDSA profile: Since the cryptographic mechanism is changed by this certificate, the domain parameters MUST be included in this certificate.

ID	LINK_CERT_13
Purpose	**For MIG profile only:** This certificate is a link certificate, which defines a new cryptographic mechanism to be used by chip.
Version	1.11
Referred by	Test case EAC2_ISO7816_N_1
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 C3 **5F 25** 06 *gg* **5F 24** 06 *hh* *optional 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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)

	vv is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETEST_LINKDE011
	Certificate Holder Reference	DETESTLINKDE013
	Certificate Holder Authorization	CVCA, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 7 months
	Certificate expiration date	$CVCA_{exp}$ + 10 month
	Public Key reference	Public key of key pair CVCA_KEY_13
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_11
	Certificate Extension	As defined by CVCA

2.4.13.2 DV_CERT_13

ID	DV_CERT_13	
Purpose	**For MIG profile only:** This certificate is a domestic DV certificate, which was issued by the new CVCA.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_N_1	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETEST_LINKDE013
	Certificate Holder Reference	DETESTDVDE013

	Certificate Holder Authorization	domestic DV, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 7 months
	Certificate expiration date	$CVCA_{exp}$ + 8 months
	Public Key reference	Public key of key pair DV_KEY_13
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_13

2.4.13.3 IS_CERT_13

ID	IS_CERT_13	
Purpose	**For MIG profile only:** This certificate is a regular IS certificate, which is issued by the DV_CERT_13.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_N_1	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE013
	Certificate Holder Reference	DETESTISDE013
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{exp}$ + 7 months
	Certificate expiration date	$CVCA_{exp}$ + 8 months
	Public Key reference	Public key of key pair IS_KEY_13
	Signing Key reference	Signed with the private key of key pair DV_KEY_13

2.4.14 Certificate Set 14

The certificate set follows a certification scheme where the DV and IS contain public key information from a generated key whose lengths are shorter than the CVCA key length.

2.4.14.1 DV_CERT_14a

ID	DV_CERT_14a	
Purpose	This certificate is a regular domestic DV certificate which is issued by the CVCA.	
Version	1.11	
Referred by	Test case EAC2_ISO7816_J_52	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE014
	Certificate Holder Authorization	domestic DV, DG 3, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_14a
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.14.2 DV_CERT_14b

ID	DV_CERT_14b

Purpose	Certificate with a wrong (short) public 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	1.11	
Referred by	Test case EAC2_ISO7816_J_51	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 01 53 01 83 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the CVCA
	Certificate Holder Reference	DETESTDVDE014
	Certificate Holder Authorization	domestic DV, DG 3, DG4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_14b
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_00

2.4.14.3 IS_CERT_14a

ID	IS_CERT_14a
Purpose	This certificate is a regular IS certificate, which is issued by the DV_CERT_14.
Version	1.11

Referred by	Test case EAC2_ISO7816_J_51	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE014
	Certificate Holder Reference	DETESTISDE014
	Certificate Holder Authorization	IS, DG 3, DG 4
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 14 days
	Public Key reference	Public key of key pair IS_KEY_14a
	Signing Key reference	Signed with the private key of key pair DV_KEY_14b

2.4.14.4 IS_CERT_14b

ID	IS_CERT_14b
Purpose	Certificate with a wrong (short) Public key. For RSA profile, same Algorithm Identifier but IS key modulus length is shorter than the DVCA's key modulus length. For ECDSA profile, same Algorithm Identifier but IS key domain parameters are different and have a shorter prime length than the DVCA's key. The hash algorithm should be adapted if necessary.
Version	1.11
Referred by	Test case EAC2_ISO7816_J_52
Content	**7F 21** *aa*

<table>
<tr><td>definition</td><td colspan="2">
7F 4E bb
5F 29 01 00
42 cc dd
7F 49 ee ff
5F 20 xx yy
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 of the certificate body object
cc is the encoded length of the Certification Authority Reference
dd is the placeholder for the Certification Authority Reference (cc bytes)
ee is the encoded length of the certificate's public key,
ff is the placeholder for the certificate's public key bytes (ee bytes),
xx is the encoded length of the Certificate Holder Reference
yy is the placeholder for the Certificate Holder Reference (xx 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)
</td></tr>
<tr><td rowspan="5">Parameter</td><td>Certification Authority Reference</td><td>DETESTDVDE014</td></tr>
<tr><td>Certificate Holder Reference</td><td>DETESTISDE014</td></tr>
<tr><td>Certificate Holder Authorization</td><td>IS, DG 3, DG 4</td></tr>
<tr><td>Certificate effective date</td><td>$CVCA_{eff}$</td></tr>
<tr><td>Certificate expiration date</td><td>$CVCA_{eff}$ + 14 days</td></tr>
<tr><td></td><td>Public Key reference</td><td>Public key of key pair IS_KEY_14b</td></tr>
<tr><td></td><td>Signing Key reference</td><td>Signed with the private key of key pair DV_KEY_14a</td></tr>
</table>

2.4.15 Certificate Set 15

Deleted in version 1.1.

2.4.16 Certificate Set 16

Deleted in version 1.1.

2.4.17 Certificate Set 17

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding eID special functions. The DV certificate permits special eID functions while the terminal certificate may restrict this access. The DV certificate is an official domestic certificate.

2.4.17.1 DV_CERT_17

ID	DV_CERT_17
Purpose	This certificate is a regular DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits access to all eID special functions. It also permits read access to DG1 for testing access permissions.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_17, Test case EAC2_ISO7816_L_18, Test case EAC2_ISO7816_L_19, Test case EAC2_ISO7816_L_20, Test case EAC2_ISO7816_L_21, Test case EAC2_ISO7816_L_22, Test case EAC2_ISO7816_L_23, Test case EAC2_ISO7816_L_24, Test case EAC2_ISO7816_L_25, Test case EAC2_ISO7816_L_26, Test case EAC2_ISO7816_L_27, Test case EAC2_ISO7816_L_28, Test case EAC2_ISO7816_M_6, Test case EAC2_ISO7816_O_9, Test case EAC2_ISO7816_O_10, Test case EAC2_ISO7816_O_11, Test case EAC2_ISO7816_O_12, Test case EAC2_ISO7816_P_15, Test case EAC2_ISO7816_P_16, Test case EAC2_ISO7816_P_17, Test case EAC2_ISO7816_P_18, Test case EAC2_ISO7816_Q_1, Test case EAC2_ISO7816_Q_2, Test case EAC2_ISO7816_Q_3, Test case EAC2_ISO7816_Q_4, Test case EAC2_ISO7816_Q_6, Test case EAC2_ISO7816_Q_7, Test case EAC2_ISO7816_Q_8, Test case EAC2_ISO7816_Q_10, Test case EAC2_ISO7816_Q_11, Test case EAC2_ISO7816_Q_12, Test case EAC2_ISO7816_Q_13, Test case EAC2_ISO7816_Q_15, Test case EAC2_ISO7816_R_1, Test case EAC2_ISO7816_R_3, Test case EAC2_ISO7816_R_5, Test case EAC2_ISO7816_R_6, Test case EAC2_ISO7816_V_1,Test case EAC2_ISO7816_V_2, Test case EAC2_ISO7816_V_3, Test case EAC2_ISO7816_V_4, Test case EAC2_ISO7816_V_5, Test case EAC2_ISO7816_V_6, Test case EAC2_ISO7816_V_7, Test case EAC2_ISO7816_V_8, Test case EAC2_ISO7816_V_9, Test case EAC2_ISO7816_V_10, Test case EAC2_ISO7816_V_11, Test case EAC2_ISO7816_V_12
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 80 40 00 01 FF **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes)

	ee is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE017
	Certificate Holder Authorization	Official domestic DV, eID-Specials (all), DG1
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_17
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.17.2 AT_CERT_17a

ID	AT_CERT_17a
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “CAN allowed”. To test read access without PIN, access to DG1 is granted.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_23, Test case EAC2_ISO7816_Q_3, Test case EAC2_ISO7816_Q_12
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 01 10 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, CAN allowed, read DG1
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.3 AT_CERT_17b

ID	AT_CERT_17b
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “PIN Management”. Special function “CAN allowed” is additionally set in order to enable an alternative PACE password for PIN management function “Activate PIN”.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_25, Test case EAC2_ISO7816_L_26, Test case EAC2_ISO7816_P_15, Test case EAC2_ISO7816_P_16, Test case EAC2_ISO7816_P_17, Test case EAC2_ISO7816_P_18, Test case EAC2_ISO7816_O_12

Content definition	**7F 21** *aa*
	  7F 4E *bb*
	  5F 29 01 00
	  42 *cc dd*
	  7F 49 *ee ff*
	  5F 20 *xx yy*
	  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 30
	  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 of the certificate body object
	cc is the encoded length of the Certification Authority Reference
	dd is the placeholder for the Certification Authority Reference (cc bytes)
	ee is the encoded length of the certificate's public key,
	ff is the placeholder for the certificate's public key bytes (ee bytes),
	xx is the encoded length of the Certificate Holder Reference
	yy is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, PIN Management, CAN allowed
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.4 AT_CERT_17c

ID	AT_CERT_17c
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “RI”.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_21, Test case EAC2_ISO7816_L_22, Test case EAC2_ISO7816_O_11, Test case EAC2_ISO7816_R_1, Test case EAC2_ISO7816_R_3, Test case EAC2_ISO7816_R_5, Test case EAC2_ISO7816_R_6

Content definition	**7F 21** *aa*
	  7F 4E *bb*
	  5F 29 01 00
	  42 *cc dd*
	  7F 49 *ee ff*
	  5F 20 *xx yy*
	  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 04
	  5F 25 06 *gg*
	  5F 24 06 *hh*
	  65 *kk* 73 L_{73} 06 09 04 00 7F 00 07 03 01 03 02 80 *ll* *mm*
	  5F 37 *ii jj*
	
	aa is the encoded combined length of certificate body and signature objects
	bb is the encoded length of the certificate body object
	cc is the encoded length of the Certification Authority Reference
	dd is the placeholder for the Certification Authority Reference (cc bytes)
	ee is the encoded length of the certificate's public key,
	ff is the placeholder for the certificate's public key bytes (ee bytes),
	xx is the encoded length of the Certificate Holder Reference
	yy is the placeholder for the Certificate Holder Reference (xx 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) *kk* is the encoded length of the certificate extension object, *ll* is the encoded length of the terminal sector hash *mm* is the placeholder for the terminal sector hash	
Parameter	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, RI
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.5 AT_CERT_17d

ID	AT_CERT_17d
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “Install Qualified Certificate”.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_18, Test case EAC2_ISO7816_L_20, Test case EAC2_ISO7816_L_22, Test case EAC2_ISO7816_L_24, Test case EAC2_ISO7816_L_27**,** Test case EAC2_ISO7816_L_28

Content definition	**7F 21** *aa*
	  7F 4E *bb*
	  5F 29 01 00
	  42 *cc dd*
	  7F 49 *ee ff*
	  5F 20 *xx yy*
	  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 80
	  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 of the certificate body object
	cc is the encoded length of the Certification Authority Reference
	dd is the placeholder for the Certification Authority Reference (cc bytes)
	ee is the encoded length of the certificate's public key,
	ff is the placeholder for the certificate's public key bytes (ee bytes),
	xx is the encoded length of the Certificate Holder Reference
	yy is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, Install Qualified Certificate
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.6 AT_CERT_17e

ID	AT_CERT_17e	
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “Install Certificate”.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_Q_4	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 40 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, Install Certificate

	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.7 AT_CERT_17f

ID	AT_CERT_17f	
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “Age Verification”.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_L_17, Test case EAC2_ISO7816_O_9, Test case EAC2_ISO7816_Q_1, Test case EAC2_ISO7816_Q_2	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 01 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, Age Verification
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17

	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.8 AT_CERT_17g

ID	AT_CERT_17g	
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “Municipality ID Check”.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_L_19, Test case EAC2_ISO7816_O_10, Test case EAC2_ISO7816_Q_6, Test case EAC2_ISO7816_Q_7, Test case EAC2_ISO7816_Q_8, Test case EAC2_ISO7816_Q_10, Test case EAC2_ISO7816_Q_11, Test case EAC2_ISO7816_Q_13, Test case EAC2_ISO7816_Q_15	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 02 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, Municipality ID Check
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.9 AT_CERT_17h

ID	AT_CERT_17h	
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “CAN allowed” and “Privileged Terminal”.	
Version	EAC2_1.1	
Referred by	Test case EAC2_ISO7816_L_37	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 18 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, CAN allowed, Privileged Terminal
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.10 AT_CERT_17i

ID	AT_CERT_17i
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special function “PSA allowed” and “Privileged Terminal” and a sector public key.
Version	EAC2_1.1

Referred by	, Test case EAC2_ISO7816_V_1,Test case EAC2_ISO7816_V_2, Test case EAC2_ISO7816_V_3, Test case EAC2_ISO7816_V_4, Test case EAC2_ISO7816_V_5, Test case EAC2_ISO7816_V_6, Test case EAC2_ISO7816_V_7, Test case EAC2_ISO7816_V_8, Test case EAC2_ISO7816_V_9, Test case EAC2_ISO7816_V_10, Test case EAC2_ISO7816_V_11, Test case EAC2_ISO7816_V_12	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 40 00 00 08 **5F 25** 06 *gg* **5F 24** 06 *hh* *6***5** *kk 73* L_{73} *06 09 04 00 7F 00 07 03 01 03 03 A0* L_{A0} *80 ll mm 81 nn oo* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *kk* is the encoded length of the certificate extension object, *ll* is the encoded length of a domain parameter ID *mm* is the placeholder for the domain parameter ID *nn* is the encoded length of a sector public key hash *oo* is the placeholder for the second sector public key hash	
Parameter	Certification Authority Reference	DETESTDVDE017
	Certificate Holder Reference	DETESTATDE017
	Certificate Holder Authorization	Terminal, PSA allowed, Privileged Terminal
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_17
	Signing Key reference	Signed with the private key of key pair DV_KEY_17

2.4.17.11 AT_CERT_17j

<table>
<tr><td>ID</td><td colspan="2">AT_CERT_17j</td></tr>
<tr><td>Purpose</td><td colspan="2">This certificate is a regular terminal certificate, which is issued by the DV_CERT_17. It encodes access rights for the eID special functions “Age Verification” and “Municipality ID Check”.</td></tr>
<tr><td>Version</td><td colspan="2">EAC2_1.1</td></tr>
<tr><td>Referred by</td><td colspan="2">Test case EAC2_ISO7816_Q_18</td></tr>
<tr><td>Content definition</td><td colspan="2">7F 21 aa
 7F 4E bb
  5F 29 01 00
  42 cc dd
  7F 49 ee ff
  5F 20 xx yy
  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 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 of the certificate body object
cc is the encoded length of the Certification Authority Reference
dd is the placeholder for the Certification Authority Reference (cc bytes)
ee is the encoded length of the certificate's public key,
ff is the placeholder for the certificate's public key bytes (ee bytes),
xx is the encoded length of the Certificate Holder Reference
yy is the placeholder for the Certificate Holder Reference (xx 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)</td></tr>
<tr><td rowspan="7">Parameter</td><td>Certification Authority Reference</td><td>DETESTDVDE017</td></tr>
<tr><td>Certificate Holder Reference</td><td>DETESTATDE017</td></tr>
<tr><td>Certificate Holder Authorization</td><td>Terminal, Age Verification, Municipality ID Check</td></tr>
<tr><td>Certificate effective date</td><td>$CVCA_{eff}$</td></tr>
<tr><td>Certificate expiration date</td><td>$CVCA_{eff}$ + 1 month</td></tr>
<tr><td>Public Key reference</td><td>Public key of key pair AT_KEY_17</td></tr>
<tr><td>Signing Key reference</td><td>Signed with the private key of key pair DV_KEY_17</td></tr>
</table>

2.4.18 Certificate Set 18

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding eID special functions. The DV certificate permits special eID functions while the terminal certificate may restrict this access. The DV certificate is a non-official certificate.

2.4.18.1 DV_CERT_18

<table>
<tr><td>ID</td><td colspan="2">DV_CERT_18</td></tr>
<tr><td>Purpose</td><td colspan="2">This certificate is a non-official DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits access to all eID special functions.</td></tr>
<tr><td>Version</td><td colspan="2">EAC2_1.0</td></tr>
<tr><td>Referred by</td><td colspan="2">Test case EAC2_ISO7816_Q_5, Test case EAC2_ISO7816_Q_9, Test case EAC2_ISO7816_Q_14, Test case EAC2_ISO7816_Q_16, Test case EAC2_ISO7816_R_8</td></tr>
<tr><td>Content definition</td><td colspan="2">7F 21 aa
  7F 4E bb
    5F 29 01 00
    42 cc dd
    7F 49 ee ff
    5F 20 xx yy
    7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 40 40 00 00 FF
    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 of the certificate body object
cc is the encoded length of the Certification Authority Reference
dd is the placeholder for the Certification Authority Reference (cc bytes)
ee is the encoded length of the certificate's public key,
ff is the placeholder for the certificate's public key bytes (ee bytes),
xx is the encoded length of the Certificate Holder Reference
yy is the placeholder for the Certificate Holder Reference (xx 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)</td></tr>
<tr><td rowspan="6">Parameter</td><td>Certification Authority Reference</td><td>As defined by the initial AT CVCA reference</td></tr>
<tr><td>Certificate Holder Reference</td><td>DETESTDVDE018</td></tr>
<tr><td>Certificate Holder Authorization</td><td>non-official DV, eID-Specials (all)</td></tr>
<tr><td>Certificate effective date</td><td>$CVCA_{eff}$</td></tr>
<tr><td>Certificate expiration date</td><td>$CVCA_{eff}$ + 1 month</td></tr>
<tr><td>Public Key reference</td><td>Public key of key pair DV_KEY_18</td></tr>
</table>

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.18.2 AT_CERT_18a

ID	AT_CERT_18a
Version	deleted in version 1.00

2.4.18.3 AT_CERT_18b

ID	AT_CERT_18b
Version	deleted in version 1.00

2.4.18.4 AT_CERT_18c

ID	AT_CERT_18c
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_18. It encodes access rights for the eID special function “RI”.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_R_8
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 04 **5F 25** 06 *gg* **5F 24** 06 *hh* **65** *kk* 73 L_{73} 06 09 04 00 7F 00 07 03 01 03 02 80 *ll mm* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *kk* is the encoded length of the certificate extension object, *ll* is the encoded length of the terminal sector hash *mm* is the placeholder for the terminal sector hash

Parameter	Certification Authority Reference	DETESTDVDE018
	Certificate Holder Reference	DETESTATDE018
	Certificate Holder Authorization	Authentication Terminal, RI
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_18
	Signing Key reference	Signed with the private key of key pair DV_KEY_18

2.4.18.5 AT_CERT_18d

ID	AT_CERT_18d
Version	deleted in version 1.00

2.4.18.6 AT_CERT_18e

ID	AT_CERT_18e
Version	deleted in version 1.00

2.4.18.7 AT_CERT_18f

ID	AT_CERT_18f
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_18. It encodes access rights for the eID special function “Age Verification”.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_Q_5
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 01 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE018
	Certificate Holder Reference	DETESTATDE018
	Certificate Holder Authorization	Terminal, Age Verification
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_18
	Signing Key reference	Signed with the private key of key pair DV_KEY_18

2.4.18.8 AT_CERT_18g

ID	AT_CERT_18g
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_18. It encodes access rights for the eID special function “Municipality ID Check”.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_Q_9, Test case EAC2_ISO7816_Q_14, Test case EAC2_ISO7816_Q_16
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 02 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTDVDE018
	Certificate Holder Reference	DETESTATDE018
	Certificate Holder Authorization	Terminal, Municipality ID Check
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_18
	Signing Key reference	Signed with the private key of key pair DV_KEY_18

2.4.19 Certificate Set 19

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding eID read access. The DV certificate permits read access to all elementary files while the terminal certificate may restrict this access. The DV certificate is an official domestic certificate.

2.4.19.1 DV_CERT_19

ID	DV_CERT_19
Purpose	This certificate is a regular DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits read access to all elementary files
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_13 Template, Test case EAC2_ISO7816_L_15 Template , Test case EAC2_ISO7816_L16 Template, Test case EAC2_ISO7816_O_5 Template
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 80 3F FF FF 10 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE019
	Certificate Holder Authorization	Official domestic DV, Read Access (all), CAN allowed
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_19
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.19.2 DV_CERT_19a

ID	DV_CERT_19a
Purpose	This certificate is a regular DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits read access to all elementary files. It is a copy of DV_CERT_19 with the exception that all RFU bits within CHAT are set to 1.
Version	EAC2_1.0
Referred by	Test case EAC2_ISO7816_L_36
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 80 BF FF FF 10 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE019
	Certificate Holder Authorization	Official domestic DV, Read Access (all), RFU=1, CAN allowed
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_19
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.19.3 AT_CERT_19_Template

ID	AT_CERT_19_template	
Purpose	This certificate defines a template of a regular terminal certificate, which is issued by the DV_CERT_19. The access rights are defined in a separate table	
Version	see Table 1	
Referred by	see Table 1	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 *<AC-DO>* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *<AC-DO>* is the access conditions data object as defined in Table 1	
Parameter	Certification Authority Reference	DETESTDVDE019
	Certificate Holder Reference	DETESTATDE019
	Certificate Holder Authorization	see Table 1, column CHA, CAN allowed

	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_19
	Signing Key reference	Signed with the private key of key pair DV_KEY_19

2.4.19.4 AT_CERT_19a to AT_CERT_19w

ID	Purpose	Version	Referred by	AC-DO	CHA
AT_CERT_19a	Read access DG1	EAC2_1.0	EAC2_ISO7816_L_13a	53 05 00 00 00 01 10	Terminal, read DG1
AT_CERT_19b	Read access DG2	EAC2_1.0	EAC2_ISO7816_L_13b	53 05 00 00 00 02 10	Terminal, read DG2
AT_CERT_19c	Read access DG3	EAC2_1.0	EAC2_ISO7816_L_13c	53 05 00 00 00 04 10	Terminal, read DG3
AT_CERT_19d	Read access DG4	EAC2_1.0	EAC2_ISO7816_L_13d	53 05 00 00 00 08 10	Terminal, read DG4
AT_CERT_19e	Read access DG5	EAC2_1.0	EAC2_ISO7816_L_13e	53 05 00 00 00 10 10	Terminal, read DG5
AT_CERT_19f	Read access DG6	EAC2_1.0	EAC2_ISO7816_L_13f	53 05 00 00 00 20 10	Terminal, read DG6
AT_CERT_19g	Read access DG7	EAC2_1.0	EAC2_ISO7816_L_13g	53 05 00 00 00 40 10	Terminal, read DG7
AT_CERT_19h	Read access DG8	EAC2_1.0	EAC2_ISO7816_L_13h	53 05 00 00 00 80 10	Terminal, read DG8
AT_CERT_19i	Read access DG9	EAC2_1.0	EAC2_ISO7816_L_13i	53 05 00 00 01 00 10	Terminal, read DG9
AT_CERT_19j	Read access DG10	EAC2_1.0	EAC2_ISO7816_L_13j	53 05 00 00 02 00 10	Terminal, read DG10
AT_CERT_19k	Read access DG11	EAC2_1.0	EAC2_ISO7816_L_13k	53 05 00 00 04 00 10	Terminal, read DG11
AT_CERT_19l	Read access DG12	EAC2_1.0	EAC2_ISO7816_L_13l	53 05 00 00 08 00 10	Terminal, read DG12
AT_CERT_19m	Read access DG13	EAC2_1.0	EAC2_ISO7816_L_13m	53 05 00 00 10 00 10	Terminal, read DG13
AT_CERT_19n	Read access DG14	EAC2_1.0	EAC2_ISO7816_L_13n	53 05 00 00 20 00 10	Terminal, read DG14
AT_CERT_19o	Read access DG15	EAC2_1.0	EAC2_ISO7816_L_13o	53 05 00 00 40 00 10	Terminal, read DG15
AT_CERT_19p	Read access DG16	EAC2_1.0	EAC2_ISO7816_L_13p	53 05 00 00 80 00 10	Terminal, read DG16
AT_CERT_19q	Read access DG17	EAC2_1.0	EAC2_ISO7816_L_13q, EAC2_ISO7816_L_15q, EAC2_ISO7816_L_16q	53 05 00 01 00 00 10	Terminal, read DG17
AT_CERT_19r	Read access DG18	EAC2_1.0	EAC2_ISO7816_L_13r, EAC2_ISO7816_L_15r, EAC2_ISO7816_L_16r	53 05 00 02 00 00 10	Terminal, read DG18
AT_CERT_19s	Read access DG19	EAC2_1.0	EAC2_ISO7816_L_13s,	53 05 00 04 00 00 10	Terminal, read DG19

			EAC2_ISO7816_L_15s, EAC2_ISO7816_L_16s		
AT_CERT_19t	Read access DG20	EAC2_1.0	EAC2_ISO7816_L_13t, EAC2_ISO7816_L_15t, EAC2_ISO7816_L_16t	53 05 00 08 00 00 10	Terminal, read DG20
AT_CERT_19u	Read access DG21	EAC2_1.0	EAC2_ISO7816_L_13u, EAC2_ISO7816_L_15u, EAC2_ISO7816_L_16u	53 05 00 10 00 00 10	Terminal, read DG21
AT_CERT_19v	Read access DG22	EAC2_1.1	EAC2_ISO7816_L_13v, EAC2_ISO7816_L_15v, EAC2_ISO7816_L_16v	53 05 00 20 00 00 10	Terminal, read DG22
AT_CERT_19w	Read access DG1	EAC2_1.0	EAC2_ISO7816_L_36	53 05 00 80 00 01 10	Terminal, read DG1, RFU=1

Table 1: Authorization of Authentication Terminals, Certificate issued by DV_CERT_19

2.4.20 Certificate Set 20

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding eID read access. The DV certificate permits read access to all elementary files while the terminal certificate may restrict this access. The DV certificate is a non-official certificate.

2.4.20.1 DV_CERT_20

ID	DV_CERT_20	
Purpose	This certificate is a non-official DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits read access to all elementary files	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_L_14 Template, Test case EAC2_ISO7816_O_6 Template	

Content definition	**7F 21** *aa*
	  7F 4E *bb*
	  5F 29 01 00
	  42 *cc dd*
	  7F 49 *ee ff*
	  5F 20 *xx yy*
	  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 40 3F FF FF 00
	  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 of the certificate body object
	cc is the encoded length of the Certification Authority Reference
	dd is the placeholder for the Certification Authority Reference (cc bytes)
	ee is the encoded length of the certificate's public key,
	ff is the placeholder for the certificate's public key bytes (ee bytes),
	xx is the encoded length of the Certificate Holder Reference
	yy is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE020
	Certificate Holder Authorization	non-official DV
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_20

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.20.2 AT_CERT_20_Template

ID	AT_CERT_20a	
Purpose	This certificate defines a template of a regular terminal certificate, which is issued by the DV_CERT_20. The access rights are defined in a separate table.	
Version	see Table 2	
Referred by	see Table 2	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 *<AC-DO>* **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) <*AC-DO*> is the access conditions data object as defined in Table 2	
Parameter	Certification Authority Reference	DETESTDVDE020
	Certificate Holder Reference	DETESTATDE020
	Certificate Holder Authorization	See Table 2, column CHA
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_20
	Signing Key reference	Signed with the private key of key pair DV_KEY_20

2.4.20.3 AT_CERT_20a to AT_CERT_20v

ID	Purpose	Version	Referred by	AC-DO	CHA
AT_CERT_20a	Read access DG1	EAC2_1.0	EAC2_ISO7816_L_14a	53 05 00 00 00 01 00	Terminal, read DG1
AT_CERT_20b	Read access DG2	EAC2_1.0	EAC2_ISO7816_L_14b	53 05 00 00 00 02 00	Terminal, read DG2
AT_CERT_20c	Read access DG3	EAC2_1.0	EAC2_ISO7816_L_14c	53 05 00 00 00 04 00	Terminal, read DG3
AT_CERT_20d	Read access DG4	EAC2_1.0	EAC2_ISO7816_L_14d	53 05 00 00 00 08 00	Terminal, read DG4
AT_CERT_20e	Read access DG5	EAC2_1.0	EAC2_ISO7816_L_14e	53 05 00 00 00 10 00	Terminal, read DG5
AT_CERT_20f	Read access DG6	EAC2_1.0	EAC2_ISO7816_L_14f	53 05 00 00 00 20 00	Terminal, read DG6
AT_CERT_20g	Read access DG7	EAC2_1.0	EAC2_ISO7816_L_14g	53 05 00 00 00 40 00	Terminal, read DG7
AT_CERT_20h	Read access DG8	EAC2_1.0	EAC2_ISO7816_L_14h	53 05 00 00 00 80 00	Terminal, read DG8
AT_CERT_20i	Read access DG9	EAC2_1.0	EAC2_ISO7816_L_14i	53 05 00 00 01 00 00	Terminal, read DG9
AT_CERT_20j	Read access DG10	EAC2_1.0	EAC2_ISO7816_L_14j	53 05 00 00 02 00 00	Terminal, read DG10
AT_CERT_20k	Read access DG11	EAC2_1.0	EAC2_ISO7816_L_14k	53 05 00 00 04 00 00	Terminal, read DG11
AT_CERT_20l	Read access DG12	EAC2_1.0	EAC2_ISO7816_L_14l	53 05 00 00 08 00 00	Terminal, read DG12
AT_CERT_20m	Read access DG13	EAC2_1.0	EAC2_ISO7816_L_14m	53 05 00 00 10 00 00	Terminal, read DG13
AT_CERT_20n	Read access DG14	EAC2_1.0	EAC2_ISO7816_L_14n	53 05 00 00 20 00 00	Terminal, read DG14
AT_CERT_20o	Read access DG15	EAC2_1.0	EAC2_ISO7816_L_14o	53 05 00 00 40 00 00	Terminal, read DG15
AT_CERT_20p	Read access DG16	EAC2_1.0	EAC2_ISO7816_L_14p	53 05 00 00 80 00 00	Terminal, read DG16
AT_CERT_20q	Read access DG17	EAC2_1.0	EAC2_ISO7816_L_14q	53 05 00 01 00 00 00	Terminal, read DG17
AT_CERT_20r	Read access DG18	EAC2_1.0	EAC2_ISO7816_L_14r	53 05 00 02 00 00 00	Terminal, read DG18
AT_CERT_20s	Read access DG19	EAC2_1.0	EAC2_ISO7816_L_14s	53 05 00 04 00 00 00	Terminal, read DG19
AT_CERT_20t	Read access DG20	EAC2_1.0	EAC2_ISO7816_L_14t	53 05 00 08 00 00 00	Terminal, read DG20
AT_CERT_20u	Read access DG21	EAC2_1.0	EAC2_ISO7816_L_14u	53 05 00 10 00 00 00	Terminal, read DG21
AT_CERT_20v	Read access DG22	EAC2_1.1	EAC2_ISO7816_L_14v	53 05 00 20 00 00 00	Terminal, read DG22

Table 2: Authorization of Authentication Terminals, Certificate issued by DV_CERT_20

2.4.21 Certificate Set 21

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding eID write access. The DV certificate permits write access to all elementary files while the terminal certificate may restrict this access. The DV certificate is an official domestic certificate.

2.4.21.1 DV_CERT_21

ID	DV_CERT_21	
Purpose	This certificate is a regular DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits write access to all elementary files	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_L_15 Template, Test case EAC2_ISO7816_O_7 Template	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 BF 00 00 00 10 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE021
	Certificate Holder Authorization	Official domestic DV, write access (all), CAN allowed
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_21

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.21.2 AT_CERT_21_Template

ID	AT_CERT_21_template	
Purpose	This certificate defines a template of a regular terminal certificate, which is issued by the DV_CERT_21. The access rights are defines in a separate table	
Version	See Table 3	
Referred by	See Table 3	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 **<***AC-DO***>** **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) <*AC-DO*> are the access conditions as defined in Table 3	
Parameter	Certification Authority Reference	DETESTDVDE021
	Certificate Holder Reference	DETESTATDE021
	Certificate Holder Authorization	See Table 3, column CHA, CAN allowed
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_21
	Signing Key reference	Signed with the private key of key pair DV_KEY_21

2.4.21.3 AT_CERT_21a to AT_CERT_21f

ID	Purpose	Version	Referred by	AC-DO	CHA
AT_CERT_21a	R/W access DG17	EAC2_1.03	EAC2_ISO7816_L_15a	53 05 20 00 00 00 10	Terminal, r/w DG17
AT_CERT_21b	R/W access DG18	EAC2_1.03	EAC2_ISO7816_L_15b	53 05 10 00 00 00 10	Terminal, r/w DG18
AT_CERT_21c	R/W access DG19	EAC2_1.03	EAC2_ISO7816_L_15c	53 05 08 00 00 00 10	Terminal, r/w DG19
AT_CERT_21d	R/W access DG20	EAC2_1.03	EAC2_ISO7816_L_15d	53 05 04 00 00 00 10	Terminal, r/w DG20
AT_CERT_21e	R/W access DG21	EAC2_1.03	EAC2_ISO7816_L_15e	53 05 02 00 00 00 10	Terminal, r/w DG21
AT_CERT_21f	R/W access DG22	EAC2_1.1	EAC2_ISO7816_L_15f	53 05 01 0 00 00 10	Terminal, r/w DG22

Table 3: Authorization of Authentication Terminals, Certificate issued by DV_CERT_21

2.4.22 Certificate Set 22

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding eID write access. The DV certificate permits write access to all elementary files while the terminal certificate may restrict this access. The DV certificate is a non-official certificate.

2.4.22.1 DV_CERT_22

ID	DV_CERT_22	
Purpose	This certificate is a non-official DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits write access to all elementary files	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_L_16 Template, Test case EAC2_ISO7816_O_8 Template	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 7F 00 00 00 00 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE022
	Certificate Holder Authorization	non-official DV, write access (all)
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_22

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.22.2 AT_CERT_22_Template

ID	AT_CERT_22a	
Purpose	This certificate defines a template of a regular terminal certificate, which is issued by the DV_CERT_22. The access rights are defined in a separate table.	
Version	See Table 4	
Referred by	See Table 4	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 **<***AC-DO***>** **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) <*AC-DO*> are the access conditions as defined in Table 4	
Parameter	Certification Authority Reference	DETESTDVDE022
	Certificate Holder Reference	DETESTATDE022
	Certificate Holder Authorization	Table 4, column CHA
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_22
	Signing Key reference	Signed with the private key of key pair DV_KEY_22

2.4.22.3 AT_CERT_22a to AT_CERT_22f

ID	Purpose	Version	Referred by	AC-DO	CHA
AT_CERT_22a	R/W access DG17	EAC2_1.03	EAC2_ISO7816_L_16a	53 05 20 00 00 00 00	Terminal, r/w DG17
AT_CERT_22b	R/W access DG18	EAC2_1.03	EAC2_ISO7816_L_16b	53 05 10 00 00 00 00	Terminal, r/w DG18
AT_CERT_22c	R/W access DG19	EAC2_1.03	EAC2_ISO7816_L_16c	53 05 08 00 00 00 00	Terminal, r/w DG19
AT_CERT_22d	R/W access DG20	EAC2_1.03	EAC2_ISO7816_L_16d	53 05 04 00 00 00 00	Terminal, r/w DG20
AT_CERT_22e	R/W access DG21	EAC2_1.03	EAC2_ISO7816_L_16e	53 05 02 00 00 00 00	Terminal, r/w DG21
AT_CERT_22f	R/W access DG22	EAC2_1.1	EAC2_ISO7816_L_16f	53 05 01 00 00 00 00	Terminal, r/w DG22

Table 4: Authorization of Authentication Terminals, Certificate issued by DV_CERT_22

2.4.23 Certificate Set 23

This certificate set defines a link certificate used for the tests about the trust point update mechanism.

2.4.23.1 LINK_CERT_23a

<table>
<tr><td>ID</td><td colspan="2">LINK_CERT_23a</td></tr>
<tr><td>Purpose</td><td colspan="2">This certificate is a link certificate, which validity period starts one day before the original CVCA certificate expires.</td></tr>
<tr><td>Version</td><td colspan="2">EAC2_1.0</td></tr>
<tr><td>Referred by</td><td colspan="2">Test case EAC2_ISO7816_M_7</td></tr>
<tr><td>Content definition</td><td colspan="2">7F 21 aa
  7F 4E bb
    5F 29 01 00
    42 cc dd
    7F 49 ee ff
    5F 20 xx yy
    7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 FF 7F FF FF FF
    5F 25 06 gg
    5F 24 06 hh
    optional: 65 vv ww
  5F 37 ii jj

aa is the encoded combined length of certificate body and signature objects
bb is the encoded length of the certificate body object
cc is the encoded length of the Certification Authority Reference
dd is the placeholder for the Certification Authority Reference (cc bytes)
ee is the encoded length of the certificate's public key,
ff is the placeholder for the certificate's public key bytes (ee bytes),
xx is the encoded length of the Certificate Holder Reference
yy is the placeholder for the Certificate Holder Reference (xx 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)
vv is the encoded length of the certificate extension,
ww is the placeholder for the certificate extension (vv bytes)</td></tr>
<tr><td rowspan="6">Parameter</td><td>Certification Authority Reference</td><td>As defined by the initial AT CVCA reference</td></tr>
<tr><td>Certificate Holder Reference</td><td>DETESTLINKDE23A</td></tr>
<tr><td>Certificate Holder Authorization</td><td>CVCA, read access to all DG, all eID functions</td></tr>
<tr><td>Certificate effective date</td><td>$CVCA_{exp}$ - 1 day</td></tr>
<tr><td>Certificate expiration date</td><td>$CVCA_{exp}$ + 3 month</td></tr>
<tr><td>Public Key reference</td><td>Public key of key pair AT_CVCA_KEY_23a</td></tr>
</table>

	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17
	Certificate Extension	As defined by CVCA

2.4.23.2 LINK_CERT_23b

ID	LINK_CERT_23b	
Purpose	This certificate is a link certificate, which validity period starts one month before the previous CVCA certificate expires.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_M_7	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 FF 7F FF FF FF **5F 25** 06 *gg* **5F 24** 06 *hh* *optional: 65 vv ww* **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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) *vv* is the encoded length of the certificate extension, *ww* is the placeholder for the certificate extension (vv bytes)	
Parameter	Certification Authority Reference	DETESTLINKDE23A
	Certificate Holder Reference	DETESTLINKDE23B
	Certificate Holder Authorization	CVCA, read access to all DG, all eID functions
	Certificate effective date	$CVCA_{exp}$ + 2 month
	Certificate expiration date	$CVCA_{exp}$ + 5 month
	Public Key reference	Public key of key pair AT_CVCA_KEY_23b

	Signing Key reference	Signed with the private key of key pair AT_CVCA_KEY_23a
	Certificate Extension	As defined by CVCA

2.4.23.3 DV_CERT_23

ID	DV_CERT_23	
Purpose	This certificate is a domestic DV certificate, which was issued by the previous AT CVCA.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_M_7	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 80 3F FF FF 00 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	DETESTLINKDE23B
	Certificate Holder Reference	DETESTDVDE023
	Certificate Holder Authorization	domestic DV, read access all DGs
	Certificate effective date	$CVCA_{exp}$ + 4 month
	Certificate expiration date	$CVCA_{exp}$ + 5 month
	Public Key reference	Public key of key pair DV_KEY_23
	Signing Key reference	Signed with the private key of key pair AT_CVCA_KEY_23b

2.4.24 Certificate Set 24

This certificate set consists of a regular certificate chain (DV -> AT) which is used for the tests regarding Restricted Identification. The DV certificate permits special eID functions while the terminal certificate may restrict this access. The DV certificate is an official domestic certificate.

2.4.24.1 DV_CERT_24

ID	DV_CERT_24	
Purpose	This certificate is a official DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits RI special function.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_R_10, Test case EAC2_ISO7816_R_12	
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 80 00 00 00 04 **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 of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE024
	Certificate Holder Authorization	official DV, eID-Special RI
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_24
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.24.2 AT_CERT_24

ID	AT_CERT_24	
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_24. It encodes access rights for the eID special function “RI” and two sector public keys.	
Version	EAC2_1.0	
Referred by	Test case EAC2_ISO7816_R_10, Test case EAC2_ISO7816_R_12	

Content definition	**7F 21** *aa*
	  7F 4E *bb*
	  5F 29 01 00
	  42 *cc dd*
	  7F 49 *ee ff*
	  5F 20 *xx yy*
	  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 00 00 00 04
	  5F 25 06 *gg*
	  5F 24 06 *hh*
	  65 *kk* 73 L_{73} 06 09 04 00 7F 00 07 03 01 03 02 80 *ll mm* 81 *ll nn*
	  5F 37 *ii jj*
	
	aa is the encoded combined length of certificate body and signature objects
	bb is the encoded length of the certificate body object
	cc is the encoded length of the Certification Authority Reference
	dd is the placeholder for the Certification Authority Reference (cc bytes)
	ee is the encoded length of the certificate's public key,
	ff is the placeholder for the certificate's public key bytes (ee bytes),
	xx is the encoded length of the Certificate Holder Reference
	yy is the placeholder for the Certificate Holder Reference (xx 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)
	kk is the encoded length of the certificate extension object,
	ll is the encoded length of a sector public key hash
	mm is the placeholder for the first sector public key hash
	nn is the placeholder for the second sector public key hash	

Parameter	Certification Authority Reference	DETESTDVDE024
	Certificate Holder Reference	DETESTATDE024
	Certificate Holder Authorization	Terminal, RI
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_24
	Signing Key reference	Signed with the private key of key pair DV_KEY_24

2.4.25 Certificate Set 25

Deleted in version 1.00 RC

2.4.26 Certificate Set 26

Deleted in version 1.00 RC

2.4.27 Certificate Set 27

Deleted in version 1.1.

2.4.28 Certificate Set 28

Deleted in version 1.1.

2.4.29 Certificate Set 29

Deleted in version 1.1.

2.4.30 Certificate Set 30

The certificate set follows a certification scheme where the DV permits access to compare data groups. The right to compare data groups is encoded in additional Authorization Extensions.

2.4.30.1 DV_CERT_30

ID	DV_CERT_30
Purpose	This certificate is a regular DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits access to all eID special functions. It also permits compare access to DG1 to DG22 for testing compare permissions in Authorization Extensions.
Version	EAC2_1.1
Referred by	Test case EAC2_ISO7816_U_1_Template, Test case EAC2_ISO7816_U_2_Template, Test case EAC2_ISO7816_U_3_Template, Test case EAC2_ISO7816_U_4_Template
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 80 7F FF FF FF **5F 25** 06 *gg* **5F 24** 06 *hh* **65** *kk* 73 *ll* 06 0A 04 00 7F 00 07 03 01 02 02 01 80

	06 00 00 00 3F FF FF **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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, *kk* is the encoded length of the certificate extension object *ll* is the encoded length of the Discretionary Data Template *jj* is the placeholder for the certificates signature (ii bytes)	
Parameter	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE030
	Certificate Holder Authorization	Official domestic DV, eID-Specials (all)
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_30
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.30.2 AT_CERT_30_Template

ID	AT_CERT_30_template
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_30. It encodes access rights for the eID special function “Compare DGx” in Authorization Extensions.
Version	EAC2_1.1
Referred by	Test case EAC2_ISO7816_U_1_Template, Test case EAC2_ISO7816_U_2_Template, Test case EAC2_ISO7816_U_3_Template, Test case EAC2_ISO7816_U_4_Template
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 7F FF FF FF **5F 25** 06 *gg* **5F 24** 06 *hh* **65** *kk* 73 *ll* 06 0A 04 00 7F 00 07 03 01 02 02 01

	<Authorization Extension> **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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 *kk* is the encoded length of the certificate extension object *ll* is the encoded length of the Discretionary Data Object *ii* is the encoded length of the certificates signature object, *jj* is the placeholder for the certificates signature (ii bytes)	
Parameter	Certification Authority Reference	DETESTDVDE030
	Certificate Holder Reference	DETESTATDE030
	Certificate Holder Authorization	Terminal,
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_30
	Signing Key reference	Signed with the private key of key pair DV_KEY_30

2.4.30.3 AT_CERT_30a to AT_CERT_30w

Cert-ID	Version	Access Right	Authorization Extension
AT_CERT_30a	EAC2_1.1	Compare DG1	80 06 00 00 00 00 00 01
AT_CERT_30b	EAC2_1.1	Compare DG2	80 06 00 00 00 00 00 02
AT_CERT_30c	EAC2_1.1	Compare DG3	80 06 00 00 00 00 00 04
AT_CERT_30d	EAC2_1.1	Compare DG4	80 06 00 00 00 00 00 08
AT_CERT_30e	EAC2_1.1	Compare DG5	80 06 00 00 00 00 00 10
AT_CERT_30f	EAC2_1.1	Compare DG6	80 06 00 00 00 00 00 20
AT_CERT_30g	EAC2_1.1	Compare DG7	80 06 00 00 00 00 00 40
AT_CERT_30h	EAC2_1.1	Compare DG8	80 06 00 00 00 00 00 80
AT_CERT_30i	EAC2_1.1	Compare DG9	80 06 00 00 00 00 01 00
AT_CERT_30j	EAC2_1.1	Compare DG10	80 06 00 00 00 00 02 00

AT_CERT_30k	EAC2_1.1	Compare DG11	80 06 00 00 00 00 04 00
AT_CERT_30l	EAC2_1.1	Compare DG12	80 06 00 00 00 00 08 00
AT_CERT_30m	EAC2_1.1	Compare DG13	80 06 00 00 00 00 10 00
AT_CERT_30n	EAC2_1.1	Compare DG14	80 06 00 00 00 00 20 00
AT_CERT_30o	EAC2_1.1	Compare DG15	80 06 00 00 00 00 40 00
AT_CERT_30p	EAC2_1.1	Compare DG16	80 06 00 00 00 00 80 00
AT_CERT_30q	EAC2_1.1	Compare DG17	80 06 00 00 00 01 00 00
AT_CERT_30r	EAC2_1.1	Compare DG18	80 06 00 00 00 02 00 00
AT_CERT_30s	EAC2_1.1	Compare DG19	80 06 00 00 00 04 00 00
AT_CERT_30t	EAC2_1.1	Compare DG20	80 06 00 00 00 08 00 00
AT_CERT_30u	EAC2_1.1	Compare DG21	80 06 00 00 00 10 00 00
AT_CERT_30v	EAC2_1.1	Compare DG22	80 06 00 00 00 20 00 00
AT_CERT_30w	EAC2_1.1	No Compare	80 06 00 00 00 00 00 00

Table 5: Authorization Extension of AT_CERT_30

2.4.30.4 AT_CERT_30x

ID	AT_CERT_30x
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_30. It encodes access rights for the eID special function “Compare Document Validity”, “Compare Municipality ID” and “Compare Date of Birth” in Authorization Extensions.
Version	EAC2_1.1
Referred by	Test case EAC2_ISO7816_U_9, Test case EAC2_ISO7816_U_10, Test case EAC2_ISO7816_U_11, Test case EAC2_ISO7816_U_12
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 7F FF FF FF **5F 25** 06 *gg* **5F 24** 06 *hh* **65** *kk* 73 *ll* 06 0A 04 00 7F 00 07 03 01 02 02 01 80 06 00 00 00 02 00 84 **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects

	bb is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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 *kk* is the encoded length of the certificate extension object *ll* is the encoded length of the Discretionary Data Object *ii* is the encoded length of the certificates signature object, *jj* is the placeholder for the certificates signature (ii bytes)	
Parameter	Certification Authority Reference	DETESTDVDE030
	Certificate Holder Reference	DETESTATDE030
	Certificate Holder Authorization	Terminal
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_30
	Signing Key reference	Signed with the private key of key pair DV_KEY_30

2.4.31 Certificate Set 31

The certificate set follows a certification scheme where the DV permits access to compare data groups.

2.4.31.1 DV_CERT_31

ID	DV_CERT_31
Purpose	This certificate is a non-official DV certificate, which validity period starts at the effective date of the CVCA and expires after one month. The certificate permits access to all eID special functions. It also permits compare access to DG1 to DG22 for testing compare permissions in Authorization Extensions.
Version	EAC2_1.1
Referred by	Test case EAC2_ISO7816_U_5_Template

Content definition	**7F 21** *aa*
	  7F 4E *bb*
	  5F 29 01 00
	  42 *cc dd*
	  7F 49 *ee ff*
	  5F 20 *xx yy*
	  7F 4C 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 40 7F FF FF FF
	  5F 25 06 *gg*

	5F 24 06 *hh* **65** *kk* 73 *ll* 06 0A 04 00 7F 00 07 03 01 02 02 01 80 06 00 00 00 3F FF FF **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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 *kk* is the encoded length of the certificate extension object *ll* is the encoded length of the Discretionary Data Object *ii* is the encoded length of the certificates signature object, *jj* is the placeholder for the certificates signature (ii bytes)	
Parameter	Certification Authority Reference	As defined by the initial AT CVCA reference
	Certificate Holder Reference	DETESTDVDE030
	Certificate Holder Authorization	Non-official domestic DV, eID-Specials (all)
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair DV_KEY_31
	Signing Key reference	Signed with the private key of key pair CVCA_KEY_17

2.4.31.2 AT_CERT_31_Template

ID	AT_CERT_31_template
Purpose	This certificate is a regular terminal certificate, which is issued by the DV_CERT_31. It encodes access rights for the eID special function “Compare DGx” in Authorization Extensions.
Version	EAC2_1.1
Referred by	Test case EAC2_ISO7816_U_5_Template
Content definition	**7F 21** *aa* **7F 4E** *bb* **5F 29** 01 00 **42** *cc dd* **7F 49** *ee ff* **5F 20** *xx yy* **7F 4C** 0E 06 09 04 00 7F 00 07 03 01 02 02 53 05 00 7F FF FF FF **5F 25** 06 *gg* **5F 24** 06 *hh*

	65 *kk* 73 *ll* 06 0A 04 00 7F 00 07 03 01 02 02 01 <Authorization Extension> **5F 37** *ii jj* *aa* is the encoded combined length of certificate body and signature objects *bb* is the encoded length of the certificate body object *cc* is the encoded length of the Certification Authority Reference *dd* is the placeholder for the Certification Authority Reference (cc bytes) *ee* is the encoded length of the certificate's public key, *ff* is the placeholder for the certificate's public key bytes (ee bytes), *xx* is the encoded length of the Certificate Holder Reference *yy* is the placeholder for the Certificate Holder Reference (xx 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 *kk* is the encoded length of the certificate extension object *ll* is the encoded length of the Discretionary Data Object *ii* is the encoded length of the certificates signature object, *jj* is the placeholder for the certificates signature (ii bytes)	
Parameter	Certification Authority Reference	DETESTDVDE030
	Certificate Holder Reference	DETESTATDE030
	Certificate Holder Authorization	Terminal, Compare DGx
	Certificate effective date	$CVCA_{eff}$
	Certificate expiration date	$CVCA_{eff}$ + 1 month
	Public Key reference	Public key of key pair AT_KEY_30
	Signing Key reference	Signed with the private key of key pair DV_KEY_30

2.4.31.3 AT_CERT_31a to AT_CERT_31w

Cert-ID	Version	Access Right	Authorization Extension
AT_CERT_31a	EAC2_1.1	Compare DG1	80 06 00 00 00 00 00 01
AT_CERT_31b	EAC2_1.1	Compare DG2	80 06 00 00 00 00 00 02
AT_CERT_31c	EAC2_1.1	Compare DG3	80 06 00 00 00 00 00 04
AT_CERT_31d	EAC2_1.1	Compare DG4	80 06 00 00 00 00 00 08
AT_CERT_31e	EAC2_1.1	Compare DG5	80 06 00 00 00 00 00 10
AT_CERT_31f	EAC2_1.1	Compare DG6	80 06 00 00 00 00 00 20
AT_CERT_31g	EAC2_1.1	Compare DG7	80 06 00 00 00 00 00 40
AT_CERT_31h	EAC2_1.1	Compare DG8	80 06 00 00 00 00 00 80
AT_CERT_31i	EAC2_1.1	Compare DG9	80 06 00 00 00 00 01 00
AT_CERT_31j	EAC2_1.1	Compare DG10	80 06 00 00 00 00 02 00

AT_CERT_31k	EAC2_1.1	Compare DG11	80 06 00 00 00 00 04 00
AT_CERT_31l	EAC2_1.1	Compare DG12	80 06 00 00 00 00 08 00
AT_CERT_31m	EAC2_1.1	Compare DG13	80 06 00 00 00 00 10 00
AT_CERT_31n	EAC2_1.1	Compare DG14	80 06 00 00 00 00 20 00
AT_CERT_31o	EAC2_1.1	Compare DG15	80 06 00 00 00 00 40 00
AT_CERT_31p	EAC2_1.1	Compare DG16	80 06 00 00 00 00 80 00
AT_CERT_31q	EAC2_1.1	Compare DG17	80 06 00 00 00 01 00 00
AT_CERT_31r	EAC2_1.1	Compare DG18	80 06 00 00 00 02 00 00
AT_CERT_31s	EAC2_1.1	Compare DG19	80 06 00 00 00 04 00 00
AT_CERT_31t	EAC2_1.1	Compare DG20	80 06 00 00 00 08 00 00
AT_CERT_31u	EAC2_1.1	Compare DG21	80 06 00 00 00 10 00 00
AT_CERT_31v	EAC2_1.1	Compare DG22	80 06 00 00 00 20 00 00
AT_CERT_31w	EAC2_1.1	No Compare	80 06 00 00 00 00 00 00

Table 6: Authorization Extension of AT_CERT_31

3 Tests for layer 6 (ISO 7816)

This chapter defines the additional tests required for the extended command set used by the extended access control mechanisms.

3.1 Test case notation

The test cases defined below specify a set of command APDU which have to be sent to the test sample. While same parts of these APDUs are fixed, other elements have variable values which cannot be defined in general. The variable parts are marked by placeholder values which have to be replaced by the actual values. The following placeholders commonly used and therefore defined here in a global manner. All other placeholders are defined within the corresponding test case definition.

Placeholder	Definition
<Lc>	The length byte containing the length of the APDU command data.
<Le>	The length byte containing the length of the requested response data. Depending on the size of <Lc> the <Le> element must consist of one or two bytes (extended length). See ISO/IEC 7816-4 5.2 "*In any command APDU comprising both L_c and L_e fields (see ISO/IEC 7816-3), short and extended length fields shall not be combined: either both of them are short, or both of them are extended*."
<Ne>	Like <Le>, but placeholder for encoding within secure messaging (Tag 97)
<L_{xy}>	The encoded length of the data object *xy*.
<Cryptogram>	The encrypted part of a Secure Messaging APDU. The data content of this cryptogram is defined in the corresponding test case definition.
<Checksum>	The cryptographic checksum which is calculated over the protected parts of the Secure Messaging command.
'|'	This is a binary OR operation. A bitwise OR takes two bit patterns of equal length and performs the logical OR operation on each pair of corresponding bits. The result in each position is 0 if both bits are 0, while otherwise the result is 1.

3.2 General requirements

3.2.1 Security Status

According to the definition in the EAC 2.0 specification [R8] the Secure Messaging session MUST be aborted if and only if a secure messaging error occurs or a plain APDU is received.

In respect to the Chip Authentication mechanism the EAC 2.0 specification contains an additional specification about the security status:

3.4.2. Security Status

If Chip Authentication Version 2 was successfully performed, Secure Messaging is restarted using the derived session keys K_{MAC} and K_{Enc}. Otherwise, Secure Messaging is continued using the previously established session keys (PACE).

3.5.2. Security Status

If the key agreement step during Chip Authentication Version 3 was successfully performed, Secure Messaging is restarted using the derived session keys K_{MAC} and K_{Enc}.

Reference 1: Security Status definition in the EAC 2.0 specification

Based on these definitions, all responses received during the test cases MUST be coded in secure messaging context unless stated different in the test case. The test setup MUST check this and MUST verify the cryptographic checksum.

3.2.2 Extended length APDUs

If the size of cryptographic keys leads to certificates that exceed the size of a standard APDU, all appropriate commands have been performed as extended length APDUs. In this case, the Lc field consists of three bytes and the corresponding Le field consists of two or three bytes.

3.2.3 Command Chaining

Command chaining is only used for the General Authenticate command. For MRTD chips support of command chaining is REQUIRED and support for command chaining MUST be indicated in the historical bytes of the ATR/ATS or in the EF.ATR. For terminals support of command chaining is REQUIRED. A terminal SHOULD test whether or not the MRTD chip supports command chaining before using this option.

3.3 Unit test EAC2_ISO7816_H – Password Authenticated Connection Establishment (PACE)

This unit covers all tests about the PACE mechanism. This mechanism establishes Secure Messaging between an MRTD chip and a terminal based on weak (short) passwords with the following advantages:

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

The complete PACE mechanism is tested including robustness tests with invalid input data.

A terminal is *unauthenticated* before successfully completing Terminal Authentication. Unauthenticated terminals may only perform PIN management operations according to the password (CAN, PIN, PUK), which is used during that process.

An Authentication Terminal with effective authorization for PIN management may perform PIN management operations after completing General Authentication Procedure.

Note: This test unit has to be performed for each PACE protocol suite specified in ICS.

3.3.1 Test case EAC2_ISO7816_H_1

Test – ID	EAC2_ISO7816_H_1
Purpose	Positive test with a valid Password Authenticated Connection Establishment process and an unauthenticated terminal using CAN password
Version	EAC2_1.02
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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 new session keys are valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 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. '90 00' within a valid SM response

3.3.2 Test case EAC2_ISO7816_H_2

Test – ID	EAC2_ISO7816_H_2
Purpose	Positive test with a valid Password Authenticated Connection Establishment process and an unauthenticated terminal using PIN password
Version	EAC2_1.02
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with PIN: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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 new session keys are valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 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. '90 00' within a valid SM response

3.3.3 Test case EAC2_ISO7816_H_3

Test – ID	EAC2_ISO7816_H_3
Purpose	Positive test with a valid Password Authenticated Connection Establishment process and an unauthenticated terminal using PUK password
Version	EAC2_1.02
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with PUK password: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 04 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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 new session keys are valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 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. '90 00' within a valid SM response

3.3.4 Test case EAC2_ISO7816_H_4_Template

Test – ID	EAC2_ISO7816_H_4_Template
Purpose	Positive test with a valid Password Authenticated Connection Establishment process and a defined terminal type, i. e. submitting CHAT for Terminal Authentication
Version	see Table 7
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 <TYPE> 84 <L_{84}> <PACE domain> 7F4C <L_{7F4C}> <CHAT>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. • CHAT contains an OID and DDO. Those values and password type are defined by Table 7. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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 new session keys are valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 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> '87' <L_{87}> <Certification Authority Reference> '88' <L_{88}> <Certification Authority Reference> '90 00' (DO88 is conditional) 6. '90 00' within a valid SM response

Test case EAC2_ISO7816_H_4a to Test case EAC2_ISO7815_H_4g:

Test Case ID	Version	OID (terminal type)	DDO (relative authorization)	Pwd Type
EAC2_ISO7816_H_4a	EAC2_1.02	id-IS (Inspection System)	03	'01'
EAC2_ISO7816_H_4b	EAC2_1.02	id-IS (Inspection System)	03	'02'
EAC2_ISO7816_H_4c	EAC2_1.02	id-AT (Authentication Terminal)	3F 7F FF FF FF	'02'
EAC2_ISO7816_H_4d	EAC2_1.02	id-AT (Authentication Terminal)	3F 7F FF FF FF	'03'
EAC2_ISO7816_H_4e	EAC2_1.02	id-ST (Signature Terminal)	03	'02'
EAC2_ISO7816_H_4f	EAC2_1.02	id-ST (Signature Terminal)	03	'03'
EAC2_ISO7816_H_4g	EAC2_1.02	id-ST (Signature Terminal)	03	'04'

Table 7: Test cases EAC2_ISO7816_H_4

3.3.5 Test case EAC2_ISO7816_H_5_Template

Test – ID	EAC2_ISO7816_H_5_Template
Purpose	Negative test with a valid Password Authenticated Connection Establishment process, but Terminal Type indicated by Certificate Holder Authorization Template is not authorized to use referenced password
Version	see Table 8
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 <TYPE> 84 <L_{84}> <PACE domain> 7F4C <L_{7F4C}> <CHAT>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. • CHAT contains an OID and DDO. Those values and password type are defined by Table 8.
Expected results	1. '6A 80'

Test case EAC2_ISO7816_H_5a to Test case EAC2_ISO7815_H_5e:

Test Case ID	Version	OID (terminal type)	DDO (relative authorization)	Pwd Type
EAC2_ISO7816_H_5a	EAC2_1.0	id-IS (Inspection System)	03	'03'

EAC2_ISO7816_H_5b	EAC2_1.0	id-IS (Inspection System)	03	‘04’
EAC2_ISO7816_H_5c	EAC2_1.0	id-AT (Authentication Terminal)	3F 7F FF FF FF	‘01’
EAC2_ISO7816_H_5d	EAC2_1.0	id-AT (Authentication Terminal)	3F 7F FF FF FF	‘04’
EAC2_ISO7816_H_5e	EAC2_1.0	id-ST (Signature Terminal)	03	‘01’

Table 8: Test cases EAC2_ISO7816_H_5

3.3.6 Test case EAC2_ISO7816_H_6_Template

Test – ID	EAC2_ISO7816_H_6_Template
Purpose	Negative test with a valid Password Authenticated Connection Establishment process and a defined terminal type, i. e. submitting CHAT for Terminal Authentication, but invalid password implying invalid authentication token
Version	see Table 9
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: ’00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 <TYPE> 84 <L_{84}> <PACE domain> 7F4C <L_{7F4C}> <CHAT>’ • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. • CHAT contains an OID and DDO. Those values and password type are defined by Table 9. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: ’10 86 00 00 <Lc> 7C 00 <Le>’ 3. Send the given General Authenticate APDU to the eID Card 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>’ Use INVALID authentication token
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. ’63 00’, ’63 CX’ or checking error

Test case EAC2_ISO7816_H_6a to Test case EAC2_ISO7815_H_6g:

Test Case ID	Version	OID (terminal type)	DDO (relative authorization)	Pwd Type
EAC2_ISO7816_H_6a	EAC2_1.02	id-IS (Inspection System)	03	‘01’
EAC2_ISO7816_H_6b	EAC2_1.02	id-IS (Inspection System)	03	‘02’
EAC2_ISO7816_H_6c	EAC2_1.02	id-AT (Authentication Terminal)	3F 7F FF FF FF	‘02’
EAC2_ISO7816_H_6d	EAC2_1.02	id-AT (Authentication Terminal)	3F 7F FF FF FF	‘03’
EAC2_ISO7816_H_6e	EAC2_1.02	id-ST (Signature Terminal)	03	‘02’
EAC2_ISO7816_H_6f	EAC2_1.02	id-ST (Signature Terminal)	03	‘03’
EAC2_ISO7816_H_6g	EAC2_1.02	id-ST (Signature Terminal)	03	‘04’

Table 9: Test cases EAC2_ISO7816_H_6

3.3.7 Test case EAC2_ISO7816_H_7

Test – ID	EAC2_ISO7816_H_7
Version	deleted in version 1.00 RC

3.3.8 Test case EAC2_ISO7816_H_8

Test – ID	EAC2_ISO7816_H_8
Version	deleted in version 1.00 RC

3.3.9 Test case EAC2_ISO7816_H_9

Test – ID	EAC2_ISO7816_H_9
Version	deleted in version 1.00 RC

3.3.10 Test case EAC2_ISO7816_H_10

Test – ID	EAC2_ISO7816_H_10
Version	deleted in version 1.00 RC

3.3.11 Test case EAC2_ISO7816_H_11

Test – ID	EAC2_ISO7816_H_11
Version	deleted in version 1.00 RC

3.3.12 Test case EAC2_ISO7816_H_12

Test – ID	EAC2_ISO7816_H_12
Version	deleted in version 1.00 RC

3.3.13 Test case EAC2_ISO7816_H_13

Test – ID	EAC2_ISO7816_H_13
Version	deleted in version 1.00 RC

3.3.14 Test case EAC2_ISO7816_H_14

Test – ID	EAC2_ISO7816_H_14
Version	deleted in version 1.00 RC

3.3.15 Test case EAC2_ISO7816_H_15

Test – ID	EAC2_ISO7816_H_15
Version	deleted in version 1.00 RC

3.3.16 Test case EAC2_ISO7816_H_16

Test – ID	EAC2_ISO7816_H_16
Version	deleted in version 1.00 RC

3.3.17 Test case EAC2_ISO7816_H_17

Test – ID	EAC2_ISO7816_H_17
Version	deleted in version 1.00 RC

3.3.18 Test case EAC2_ISO7816_H_18

Test – ID	EAC2_ISO7816_H_18
Version	deleted in version 1.00 RC

3.3.19 Test case EAC2_ISO7816_H_19

Test – ID	EAC2_ISO7816_H_19
Version	deleted in version 1.00 RC

3.3.20 Test case EAC2_ISO7816_H_20

Test – ID	EAC2_ISO7816_H_20

Version	deleted in version 1.00 RC

3.3.21 Test case EAC2_ISO7816_H_21

Test – ID	EAC2_ISO7816_H_21
Version	moved in version 1.00 RC to Test case EAC2_ISO7816_P_8a

3.3.22 Test case EAC2_ISO7816_H_22

Test – ID	EAC2_ISO7816_H_22
Version	deleted in version 1.00 RC, duplicate of Test case EAC2_ISO7816_P_4

3.3.23 Test case EAC2_ISO7816_H_23

Test – ID	EAC2_ISO7816_H_23
Purpose	Test with invalid PIN/Password reference
Version	EAC2_1.0
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with invalid pin reference: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 <invalid password reference> 84 <L_{84}> <PACE domain>' • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. • The password reference (DO83) has been set to an invalid value. (see ICS, use '05' if not otherwise stated)
Expected results	1. '6A 88'

3.3.24 Test case EAC2_ISO7816_H_24

Test – ID	EAC2_ISO7816_H_24
Version	deleted in version 1.00 RC

3.3.25 Test case EAC2_ISO7816_H_25

Test – ID	EAC2_ISO7816_H_25
Version	deleted in version 1.00 RC

3.3.26 Test case EAC2_ISO7816_H_26

Test – ID	EAC2_ISO7816_H_26
Version	Deleted in version 0.99

3.3.27 Test case EAC2_ISO7816_H_27

Test – ID	EAC2_ISO7816_H_27
Version	deleted in version 1.00 RC

3.3.28 Test case EAC2_ISO7816_H_28

Test – ID	EAC2_ISO7816_H_28
Version	moved in version 1.00 RC to Test case EAC2_ISO7816_P_19

3.3.29 Test case EAC2_ISO7816_H_29

Test – ID	EAC2_ISO7816_H_29
Version	deleted in version 1.00 RC

3.3.30 Test case EAC2_ISO7816_H_30

Test – ID	EAC2_ISO7816_H_30
Version	deleted in version 1.00 RC

3.3.31 Test case EAC2_ISO7816_H_31

Test – ID	EAC2_ISO7816_H_31
Version	deleted in version 1.00 RC

3.3.32 Test case EAC2_ISO7816_H_32

Test – ID	EAC2_ISO7816_H_32
Version	deleted in version 1.00 RC

3.3.33 Test case EAC2_ISO7816_H_33

Test – ID	EAC2_ISO7816_H_33
Purpose	Test with an invalid ephemeral public key - different key size
Version	EAC2_1.02
Profile	PACE
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card:

	'00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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>' • The ephemeral public key MUST be generated with domain parameters specifying a different key size (e.g. for a 224 bit key in EF.CardAccess / EF.CardSecurity a 192 bit ephemeral key pair is created)
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. Checking error or '63 00'. Since there are invalid domain parameters used to generate the ephemeral key pair, the key agreement process MUST always fail.

3.3.34 Test case EAC2_ISO7816_H_34

Test – ID	EAC2_ISO7816_H_34
Purpose	Test with an invalid ephemeral public key - providing a (0,0) public key
Version	EAC2_1.02
Profile	PACE, ECDH
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the

	encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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>' • The ephemeral public key has both coordinates set to zero.
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. Checking error or '63 00'. Even if public key validation is not done, ECDH computation SHOULD fail with this input.

3.3.35 Test case EAC2_ISO7816_H_35

Test – ID	EAC2_ISO7816_H_35
Purpose	Test with an invalid ephemeral public key - value strictly bigger than the prime
Version	EAC2_1.02
Profile	PACE, DH
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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>' Use an ephemeral public key with a wrong value (value strictly bigger than the prime), e. g. ephemeral public key = prime p + 1
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. Checking error or ’63 00’.

3.3.36 Test case EAC2_ISO7816_H_36

Test – ID	EAC2_ISO7816_H_36
Purpose	Test with an invalid ephemeral public key – value does not belong to the curve
Version	EAC2_1.02
Profile	PACE, ECDH
Preconditions	1. None, card recently activated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card: ’00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>’ • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: ’10 86 00 00 <Lc> 7C 00 <Le>’ 3. Send the given General Authenticate APDU to the eID Card 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>’ The ephemeral public key does not belong to the curve.
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. Checking error or ’63 00’.

3.4 Unit EAC2_ISO7816_I - Chip Authentication

The chip authentication mechanism uses the manage security environment command to verify that the chip is genuine. The terminal and the eID Card generate a shared secret based on the public key data stored in EF.CardSecurity file of the document. This secret is used to derive new session keys for the continued secure messaging session. The genuineness of the MRTD chip is explicitly verified by the authentication token and 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 AT” / ”General Authentication” command pair.

EF.CardSecurity file may contain an optional key reference identifier. This is useful if the chip supports multiple keys for Chip Authentication. The MSE:Set AT command can be called either with implicit key selection if no key reference is included in EF.CardSecurity or with the explicit key reference defined in the EF.CardSecurity element. All tests in this unit SHOULD be used with implicit or explicit key reference depending on the presence of the key reference element in EF.CardSecurity.

The EF.CardSecurity may contain more than one ChipAuthenticationPublicKeyInfo. In this case, all appropriate tests must be performed for each key. The corresponding test case is only rated as a PASS if all passes are completed successfully. For test cases where the Chip Authentication mechanism is just used as precondition always the first key is used.

3.4.1 Test case EAC2_ISO7816_I_1

Test - ID	EAC2_ISO7816_I_1
Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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. Verify the returned authentication token TPICC 4. To verify the chips ability to continue the Secure Messaging with the new session keys, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. 7C <L_{7C}> '81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

	3. True 4. ’90 00’ and a valid Secure Messaging response. The returned data MUST be encoded with the NEW session keys.

3.4.2 Test case EAC2_ISO7816_I_2

Test - ID	EAC2_ISO7816_I_2
Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key, but afterward the old session keys are used.
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. ‘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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. ‘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. Verify the returned authentication token T_{PICC} 4. Instead of using the new session keys, the old session keys are used to send an arbitrary SM APDU to the chip. ‘0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00’
Expected results	1. ’90 00’ in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. 7C <L_{7C}> ’81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00’ in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 3. True 4. Checking error. The chip MUST delete the old session key and MUST NOT accept any APDUs with these session keys.

3.4.3 Test case EAC2_ISO7816_I_3

Test - ID	EAC2_ISO7816_I_3
Purpose	MSE:Set AT / General Authenticate commands with invalid ephemeral public key (different key size)
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 MUST be generated with domain parameters specifying a different key size (e.g. for a 224 bit key in EF.CardSecurity a 192 bit ephemeral key pair is created) 3. To verify that the old (PACE based) session keys can still be used, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. Checking error, or warning '63 00'. Since there are invalid domain parameters used to generate the ephemeral key pair, the key agreement process MUST always fail. 3. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.4 Test case EAC2_ISO7816_I_4

Test - ID	EAC2_ISO7816_I_4

Purpose	MSE:Set AT / General Authenticate commands with a valid ephemeral public key, but without SecureMessaging
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in EF.CardSecurity MUST have been read BEFORE to be able to generate an ephemeral key pair.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '00 22 41 A4 <Lc> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' • The private key reference MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '00 86 00 00 <Lc> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <Le>' 3. To verify that the chip has deleted the old (PACE based) session keys, an arbitrary SM APDU is send to the chip '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' or Checking error. A chip may permit the use of an unprotected MSE APDU, however, the SM channel MUST be closed as soon as an unprotected APDU is send. Therefore, the response MUST be send without SM encoding. 2. Checking error. The error code SHALL be returned as plain data without SM encoding. 3. Checking error. The error code SHALL be returned as plain data without SM encoding.

3.4.5 Test case EAC2_ISO7816_I_5

Test - ID	EAC2_ISO7816_I_5
Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key but invalid class byte
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.

Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '8C 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 class byte has been set to an invalid value of 8C. • The private key reference MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '8C 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 class byte has been set to an invalid value of 8C.
Expected results	1. Checking error. Note that the behavior of the chip regarding the Secure Messaging context is undefined. Therefore this error can be returned in plain or as an SM response. 2. Checking error. Note that the behavior of the chip regarding the Secure Messaging context is undefined. Therefore this error can be returned in plain or as an SM response.

3.4.6 Test case EAC2_ISO7816_I_6

Test - ID	EAC2_ISO7816_I_6
Purpose	MSE:Set AT / General Authenticate commands with invalid data object tag for the ephemeral public key
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file.

	2. Send the given General Authenticate APDU to the eID Card. '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 old session keys are still valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. Checking error. The error MUST be encoded in a Secure Messaging response using the OLD session keys. 3. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.7 Test case EAC2_ISO7816_I_7

Test - ID	EAC2_ISO7816_I_7
Purpose	MSE:Set AT command with wrongly appended le byte
Version	EAC2_1.0
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 01 00 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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. • The APDU has wrongly appended DO97 with an encoded Le byte. 2. To verify that the chip does not activate the new session keys, an arbitrary SM APDU using the OLD keys is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. Checking error. Note that the Secure Messaging context is not affected by this error. Therefore this error must be encoded as an SM response.

	2. ’90 00’ and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.8 Test case EAC2_ISO7816_I_8

Test - ID	EAC2_ISO7816_I_8
Purpose	MSE:Set AT / General Authenticate commands with wrongly missing le byte in GA
Version	EAC2_1.03
Profile	PACE, TA2, CA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. ‘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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. ‘0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum>’ • <Cryptogram> contains the following encrypted data objects 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> • The APDU has wrongly missing DO97. 3. To verify that the chip does not activate the new session keys, an arbitrary SM APDU using the OLD keys is send to the chip. ‘0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00’
Expected results	1. ’90 00’ in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. Checking error. Note that the behavior of the chip regarding the Secure Messaging context is undefined. Therefore this error can be returned in plain or as an SM response. 3. If Step 2 response is in plain a checking error is expected. Otherwise the expected result is ’90 00’ and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.9 Test case EAC2_ISO7816_I_9

Test - ID	EAC2_ISO7816_I_9
Purpose	MSE:Set AT / General Authenticate commands, providing a (0,0) public key to General Authenticate
Version	EAC2_1.0
Profile	PACE, TA2, CA2, ECDH
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 both coordinates set to zero. 3. To verify that the old session keys are still valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. Checking error or warning processing '63 00'. Note: Even if public key validation is not done, DH computation SHOULD fail with this input. The error MUST be encoded in a Secure Messaging response using the OLD session keys. 3. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.10 Test case EAC2_ISO7816_I_10

Test - ID	EAC2_ISO7816_I_10
Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y-

	coordinates (small x coordinate)
Version	EAC2_1.0
Profile	PACE, TA2, CA2, ECDH
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 zero bytes. (For details on q see [R6]) 3. Verify the returned authentication token T_{PICC} 4. To verify the chips ability to continue the Secure Messaging with the new session keys, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. ' 7C <L_{7C}> 81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 3. True 4. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the new session keys.

3.4.11 Test case EAC2_ISO7816_I_11

Test - ID	EAC2_ISO7816_I_11
Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y-coordinates (large x coordinate)

Version	EAC2_1.0
Profile	PACE, TA2, CA2, ECDH
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 highest bit set to 1 3. Verify the returned authentication token T_{PICC} 4. To verify the chips ability to continue the Secure Messaging with the new session keys, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. '7C <L_{7C}> 81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 3. True 4. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the new session keys.

3.4.12 Test case EAC2_ISO7816_I_12

Test - ID	EAC2_ISO7816_I_12
Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y-coordinates (small y coordinate)
Version	EAC2_1.0

Profile	PACE, TA2, CA2, ECDH
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in the EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 a y-coordinate requiring less than [$\log_{256}$ q] bytes to be represented. Pad with zero bytes. (For details on q see [R6]) 3. Verify the returned authentication token T_{PICC} 4. To verify the chips ability to continue the Secure Messaging with the new session keys, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. '7C <L_{7C}> 81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 3. True 4. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the new session keys.

3.4.13 Test case EAC2_ISO7816_I_13

Test - ID	EAC2_ISO7816_I_13
Purpose	MSE:Set AT / General Authenticate commands, test borderline cases for x- and y-coordinates (large y coordinate)
Version	EAC2_1.0
Profile	PACE, TA2, CA2, ECDH

Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 a y-coordinate having its highest bit set to 1 3. Verify the returned authentication token T_{PICC} 4. To verify the chips ability to continue the Secure Messaging with the new session keys, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. ' 7C <L_{7C}> 81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 3. True 4. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the new session keys.

3.4.14 Test case EAC2_ISO7816_I_14

Test - ID	EAC2_ISO7816_I_14
Purpose	MSE:Set AT command with an incorrect private key reference Note: The support for key references is not mandatory for the chip. This test is set optional.
Version	EAC2_1.0
Profile	PACE, TA2, CA2

Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU. This key reference MUST be different from the one potentially specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file (see ICS). 2. To verify that the old session keys are still valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. Checking error or warning processing '63 00'. The error MUST be encoded in a Secure Messaging response using the OLD session keys. 2. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.15 Test case EAC2_ISO7816_I_15

Test - ID	EAC2_ISO7816_I_15
Purpose	Check the Chip authentication failure (using DH) – wrong value (value strictly bigger than the Prime)
Version	EAC2_1.0
Profile	PACE, TA2, CA2, DH
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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 a wrong value (value strictly bigger than the Prime) ephemeral public key = prime p + 1 3. To verify that the old session keys are still valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. Checking error or warning processing '63 00'. The SW MUST be wrapped with the old session keys. Subsequent command MUST be wrapped with the old session keys. 3. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.16 Test case EAC2_ISO7816_I_16

Test - ID	EAC2_ISO7816_I_16
Purpose	Check the Chip authentication failure (using ECDH) – wrong point (value does not belong to the curve)
Version	EAC2_1.0
Profile	PACE, TA2, CA2, ECDH
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo stored in CardSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file.

	2. Send the given General Authenticate APDU to the eID Card. '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 a wrong point (value does not belong to the curve) 3. To verify that the old session keys are still valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. Checking error or warning processing '63 00'. The SW MUST be wrapped with the old session keys. Subsequent command MUST be wrapped with the old session keys. 3. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys.

3.4.17 Test case EAC2_ISO7816_I_17

Test - ID	EAC2_ISO7816_I_17
Purpose	MSE:Set AT / General Authenticate commands with correct ephemeral public key using ChipAuthenticationPublicKeyInfo encapsulated in PrivilegedTerminalInfo
Version	EAC2_1.1
Profile	PACE, TA2, CA2, CS
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication MUST have been performed (DV_CERT_1, IS_CERT_1) 3. The ChipAuthenticationPublicKeyInfo encapsulated in PrivilegedTerminalInfo stored in ChipSecurity file MUST have been read to be able to generate an ephemeral key pair. 4. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card. '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 MUST be included in the APDU if and only if it is specified in the ChipAuthenticationPublicKeyInfo structure stored in EF.CardSecurity file. 2. Send the given General Authenticate APDU to the eID Card. '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. Verify the returned authentication token TPICC 4. To verify that the old session keys are still valid, an arbitrary SM APDU is send to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 2. 7C <L_{7C}> '81 <L_{81}> <Nonce> 82 <L_{82}> <Authentication Token> 90 00' in a valid Secure Messaging response. The returned data MUST be encoded with the OLD session keys. 3. The returned authentication token is valid. 4. '90 00' and a valid Secure Messaging response. The returned data MUST be encoded with the NEW session keys.

3.5 Unit EAC2_ISO7816_J - Certificate verification

During the Terminal Authentication process the certificate chain from the trust point returned by the PACE protocol 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.

PACE mechanism is performed with CAN (IS and ST) or PIN (AT). Used Certificate Holder Authorization Template MUST match terminal type and authorization given by the certificate chain.

3.5.1 Test case EAC2_ISO7816_J_1

Test - ID	EAC2_ISO7816_J_1
Purpose	Positive test with a valid chain of CV certificates.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.2 Test case EAC2_ISO7816_J_2

Test - ID	EAC2_ISO7816_J_2
Purpose	Test with an invalid Certification Authority Reference.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '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 Certification Authority Reference> • The Certification Authority Reference returned by the PACE mechanism is changed in the last character to create an invalid reference. 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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 IS_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 checking error within 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. 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.3 Test case EAC2_ISO7816_J_3

Test - ID	EAC2_ISO7816_J_3
Purpose	Test with an invalid certificate signature.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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}> <bad certificate signature> • The signature object of the certificate has been changed in last digit to make it invalid 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '63 00' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '63 00' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.4 Test case EAC2_ISO7816_J_4

Test - ID	EAC2_ISO7816_J_4
Purpose	Test with a missing certificate signature.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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> • The certificate signature object is omitted. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '63 00' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '63 00' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.5 Test case EAC2_ISO7816_J_5

Test - ID	EAC2_ISO7816_J_5
Purpose	Test with a missing certificate body.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 5F 37 <L_{5F37}> <certificate signature> • The certificate body object is omitted. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '63 00' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '63 00' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.6 Test case EAC2_ISO7816_J_6

Test - ID	EAC2_ISO7816_J_6
Purpose	Test a DV certificate with a missing Holder Authorization.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1a. '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 contain a certificate holder authorization 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.7 Test case EAC2_ISO7816_J_7

Test - ID	EAC2_ISO7816_J_7
Purpose	Test a DV certificate with a missing effective date.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1b. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.8 Test case EAC2_ISO7816_J_8

Test - ID	EAC2_ISO7816_J_8
Purpose	Test a DV certificate with a missing expiration date.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1c. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.9 Test case EAC2_ISO7816_J_9

Test - ID	EAC2_ISO7816_J_9
Purpose	Test a DV certificate with an incorrect encoded effective date (bad BCD coding). Note: The date format verification is not mandatory for the chip. This test is set optional.
Version	EAC2_1.0
Profile	PACE, TA2, DATE
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1d.

	‘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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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’ within a valid SM response 2. Checking error or ’63 00’ within a valid SM response. 3. ’90 00’ or checking error within 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. 4. Checking error or ’63 00’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.10 Test case EAC2_ISO7816_J_10

Test - ID	EAC2_ISO7816_J_10
Purpose	Test a DV certificate with an incorrect encoded expiration date. (bad BCD coding) Note: The date format verification is not mandatory for the chip. This test is set optional.
Version	EAC2_1.0
Profile	PACE, TA2, DATE
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism.

	2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1e. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.11 Test case EAC2_ISO7816_J_11

Test - ID	EAC2_ISO7816_J_11
Purpose	Test the "Current Date" update mechanism with a new foreign IS certificate.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism.

2. Send the appropriate DV-Certificate as specified in the "Certificate Set 2" chapter as DV_CERT_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 is marked as a foreign DV-certificate.
3. Send the given MSE: Set DST APDU to the eID Card.
   '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'
   - <Cryptogram> contains the following encrypted data objects
     83 <L_{83}> <Certification Authority Reference>
   - The Certificate 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 2" chapter as IS_CERT_2a.
   '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 has an advanced effective date. Since the DV certificate was marked as a foreign one, the chip MUST 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 eID Card.
   '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'
   - <Cryptogram> contains the following encrypted data objects
     83 <L83> <Certification Authority Reference>
   - The Certification Authority Reference MUST be used as returned by the PACE mechanism.
6. Send the appropriate DV-Certificate as specified in the "Certificate Set 2" chapter as DV_CERT_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 is marked as a foreign DV-certificate.
7. Send the given MSE: Set DST APDU to the eID Card.
   '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'
   - <Cryptogram> contains the following encrypted data objects
     83 <L_{83}> <Certification Authority Reference>

	• The Certificate Holder Reference stored inside the DV-Certificate sent in step 6 has to be used. 8. Send the appropriate IS-Certificate as specified in the "Certificate Set 2" chapter as IS_CERT_2b. '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 IS-Certificate used in step 4.
Expected results	1. '90 00' within a valid SM response. 2. '90 00' within a valid SM response. 3. '90 00' within a valid SM response. 4. '90 00' within a valid SM response. 5. '90 00' within a valid SM response. 6. '90 00' within a valid SM response. 7. '90 00' within a valid SM response. 8. '90 00' within a valid SM response. This certificate MUST still be accepted since the chip MUST NOT change the current date based on the foreign IS certificate.

3.5.12 Test case EAC2_ISO7816_J_12

Test - ID	EAC2_ISO7816_J_12
Purpose	Test with a valid chain of CV certificates but without using SecureMessaging.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '00 22 81 B6 <Lc> 83 <Certification Authority Reference>' • The Certification Authority Reference MUST be used as returned by the PACE mechanism. • The APDU is send in plain without Secure Messaging 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> • The APDU is send as a valid SM APDU. • After step 2, the passport is reset and the preconditions of this test case are reestablished.

	3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. • The APDU is send as a valid SM APDU. 4. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1. '00 2A 00 BE <Lc> 7F 4E <L_{7F4E}> <body> 5F 37 <L_{5F37}> <signature>' 5. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 4 has to be used. • The APDU is send as a valid SM APDU.
Expected results	1. '90 00' or Checking error. A chip may permit the use of an unprotected MSE APDU, however, the SM channel MUST be closed as soon as an unprotected APDU is send. Therefore, the response MUST be send without SM encoding. 2. Checking error. Since the SM channel MUST have been closed in Step 1, the chip MUST return an error without SM encoding here. 3. '90 00' within a valid SM response 4. '90 00' or Checking error. A chip may permit the use of an unprotected PSO APDU, however, the SM channel MUST be closed as soon as an unprotected APDU is send. Therefore, the response MUST be send without SM encoding. 5. Checking error. Since the SM channel MUST have been closed in Step 4, the chip MUST return an error without SM encoding here.

3.5.13 Test case EAC2_ISO7816_J_13

Test - ID	EAC2_ISO7816_J_13
Purpose	Test the MSE:Set DST command with an invalid class byte.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '8C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	• <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. • The class byte is set to an invalid value. 2. If the error code in step 1 was returned in a Secure Messaging response, verify that the secure messaging session has not been aborted. If a plain error code was returned, this step is skipped. Send an arbitrary SM APDU to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. Checking error. Note that the behavior of the chip regarding the Secure Messaging context is undefined. Therefore this error can be returned in plain or as an SM response. 2. Skipped or '90 00' within a valid SM response.

3.5.14 Test case EAC2_ISO7816_J_14

Test - ID	EAC2_ISO7816_J_14
Purpose	Test the Verify Certificate command with an invalid class byte.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All commands are encoded as legally structured Secure Messaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1. '8C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • The class byte has been set to an invalid value ('8C'). 3. If the error code in step 2 was returned in a Secure Messaging response, verify that the secure messaging session has not been aborted. If a plain error code was returned, this step is skipped. Send an arbitrary SM APDU to the chip. '0C 0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'

Expected results	1. ’90 00’ within a valid SM response. 2. Checking error. Note that the behavior of the chip regarding the Secure Messaging context is undefined. Therefore this error can be returned in plain or as an SM response. 3. Skipped or ’90 00’ in a valid SM response

3.5.15 Test case EAC2_ISO7816_J_15

Test - ID	EAC2_ISO7816_J_15
Purpose	Test with an invalid certificate body tag.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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’ within a valid SM response 2. Checking error or ’63 00’ within a valid SM response.

	3. ’90 00’ or checking error within 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. 4. Checking error or ’63 00’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.16 Test case EAC2_ISO7816_J_16

Test - ID	EAC2_ISO7816_J_16
Purpose	Test with an invalid certificate signature tag.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 38 <L5F38> <certificate signature> • The certificate signature tag has been changed to ‘5F 38’ 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_CERT_1. ‘0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature>
Expected results	1. ’90 00’ within a valid SM response 2. Checking error or ’63 00’ within a valid SM response.

	3. ’90 00’ or checking error within 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. 4. Checking error or ’63 00’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.17 Test case EAC2_ISO7816_J_17

Test - ID	EAC2_ISO7816_J_17
Purpose	Test a DV certificate with an incorrect Gregorian effective date. Note: The date format verification is not mandatory for the chip. This test is set optional.
Version	EAC2_1.0
Profile	PACE, TA2, DATE
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1f. ‘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 an invalid Gregorian effective date. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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’ within a valid SM response 2. Checking error or ’63 00’ within a valid SM response. 3. ’90 00’ or checking error within 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. 4. Checking error or ’63 00’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.18 Test case EAC2_ISO7816_J_18

Test - ID	EAC2_ISO7816_J_18
Purpose	Test a DV certificate with an incorrect Gregorian expiration date. Note: The date format verification is not mandatory for the chip. This test is set optional.
Version	EAC2_1.0
Profile	PACE, TA2, DATE
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1g. ‘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 an invalid Gregorian expiration date. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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’ within a valid SM response 2. Checking error or ‘6300’ within a valid SM response. 3. ’90 00’ or checking error within 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. 4. Checking error or ‘6300’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.19 Test case EAC2_ISO7816_J_19

Test - ID	EAC2_ISO7816_J_19
Purpose	Test a DV certificate with an expiration date BEFORE the effective date.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1h. ‘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 an expiration date BEFORE the effective date. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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’ within a valid SM response 2. Checking error or ‘6300’ within a valid SM response. 3. ’90 00’ or checking error within 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. 4. Checking error or ‘6300’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.20 Test case EAC2_ISO7816_J_20

Test - ID	EAC2_ISO7816_J_20
Purpose	Test correct removal of temporary keys.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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. Reset the chip and reestablish the PACE mechanism Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response. 2. '90 00' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. The temporary key of the DV certificate MUST have been deleted during the reset. Therefore it MUST NOT be possible to verify the IS certificate based on this key.

3.5.21 Test case EAC2_ISO7816_J_21

Test - ID	EAC2_ISO7816_J_21
Purpose	Test a DV certificate with invalid combination of OID and discretionary data object in the Certificate Holder Authorization element.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1i. '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 has an invalid combination of OID (<id-AT>) and discretionary data object (structured like a relative authorization bit map for an IS) in the Certificate Holder Authorization element. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response. 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.22 Test case EAC2_ISO7816_J_22

Test - ID	EAC2_ISO7816_J_22
Purpose	Test a DV certificate invalid OID in the Public Key element.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1j. '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 has an invalid OID in the Public Key element. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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' within a valid SM response. 2. Checking error or '6300' within a valid SM response. 3. '90 00' or checking error within 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. 4. Checking error or '6300' within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.23 Test case EAC2_ISO7816_J_23

Test - ID	EAC2_ISO7816_J_23
Purpose	Test the CVCA root key selection with a wrong name (CAR) - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the MSE: Set DST APDU to initiate the certificate verification to the eID Card 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 the "Certificate Set 12" chapter as DV_CERT_12a. '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 is issued by the CVCA whose selection SHOULD have failed. • This certificate has an advanced effective date. Since the DV certificate failed, the chip MUST 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 eID Card with a correct CVCA key name (CAR). '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' • <Cryptogram> contains the encrypted Name (CAR) • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' or Checking error within a valid SM response. A chip may permit the selection of an unknown key. 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.24 Test case EAC2_ISO7816_J_24

Test - ID	EAC2_ISO7816_J_24
Purpose	Test a DV certificate with a wrong certificate body tag - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12b. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4**F** <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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.25 Test case EAC2_ISO7816_J_25

Test - ID	EAC2_ISO7816_J_25
Purpose	Test a DV certificate with a wrong certificate signature tag - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12c. '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 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.26 Test case EAC2_ISO7816_J_26

Test - ID	EAC2_ISO7816_J_26
Purpose	Test a DV certificate with a wrong certificate body length - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12d. '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}> - 1 <certificate body>

	5F 37 <L_{5F37}> <certificate signature> • The length of the certificate body is inconsistent. • This certificate has an advanced effective date. Since the DV certificate failed, the chip MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.27 Test case EAC2_ISO7816_J_27

Test - ID	EAC2_ISO7816_J_27
Purpose	Test a DV certificate with a wrong certificate signature length - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12e. '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}> - 1 <certificate signature> • The length of the certificate signature is inconsistent. • This certificate has an advanced effective date. Since the DV certificate failed, the chip MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.28 Test case EAC2_ISO7816_J_28

Test - ID	EAC2_ISO7816_J_28
Purpose	Test a DV certificate with a wrong certificate signature (Last byte increased by 1) - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism.

	2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12f. '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 + 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.29 Test case EAC2_ISO7816_J_29

Test - ID	EAC2_ISO7816_J_29
Purpose	Test a DV certificate with a wrong certificate signature (Dropping last byte of the signature) - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12g '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 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.30 Test case EAC2_ISO7816_J_30

Test - ID	EAC2_ISO7816_J_30
Purpose	Test a DV certificate with a wrong certificate signature (Signature greater than the modulus) - Current date not updated
Version	EAC2_1.0

Profile	PACE, TA2, RSA
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12o '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 signature is wrong. It is obtained by setting the signature to a value greater than the modulus. The length of the signature MUST match the length of the modulus. This certificate has an advanced effective date. Since the DV certificate failed, the chip MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.31 Test case EAC2_ISO7816_J_31

Test - ID	EAC2_ISO7816_J_31
Purpose	Test a DV certificate with a wrong certificate signature (r = 0) - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2, ECDSA
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12p '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 signature is wrong. It is obtained by filling the 'r' part of the signature with '00'. The length of 'r' still matches the size of the prime. • This certificate has an advanced effective date. Since the DV certificate failed, the chip MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response

	2. Checking error or warning processing ’63 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response

3.5.32 Test case EAC2_ISO7816_J_32

Test - ID	EAC2_ISO7816_J_32
Purpose	Test a DV certificate with a wrong certificate signature (s = 0) - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2, ECDSA
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 12” chapter as DV_CERT_12q ‘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 signature is wrong. It is obtained by filling the ‘s’ part of the signature with ‘00’. The length of ‘s’ still matches the size of the prime. • This certificate has an advanced effective date. Since the DV certificate failed, the chip MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. ’90 00’ within a valid SM response 2. Checking error or warning processing ’63 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response

3.5.33 Test case EAC2_ISO7816_J_33

Test - ID	EAC2_ISO7816_J_33
Purpose	Test a DV certificate without selecting any root key - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the appropriate DV-Certificate as specified in the “Certificate Set 12” chapter as DV_CERT_12a. ‘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> • 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 3. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.

Expected results	1. Checking error or warning processing ’63 00’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response

3.5.34 Test case EAC2_ISO7816_J_34

Test - ID	EAC2_ISO7816_J_34
Purpose	Test a DV certificate while the Public Key DO has a wrong OID field - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 12” chapter as DV_CERT_12i ‘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 Public Key DO in the certificate body contains an incorrect 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. ’90 00’ within a valid SM response 2. Checking error or warning processing ’63 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response

3.5.35 Test case EAC2_ISO7816_J_35

Test - ID	EAC2_ISO7816_J_35
Purpose	Test a DV certificate while the Public Key DO has no OID field - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 12” chapter as DV_CERT_12h. ‘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 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. Send the appropriate DV-Certificate as specified in the “Certificate Set 1” chapter as DV_CERT_1.

	'OC 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' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.36 Test case EAC2_ISO7816_J_36

Test - ID	EAC2_ISO7816_J_36
Purpose	Test a DV certificate while the Public Key DO has no Public point field - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2, ECDSA
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12j '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 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.37 Test case EAC2_ISO7816_J_37

Test - ID	EAC2_ISO7816_J_37
Purpose	Test a DV certificate while the Public Key DO has no Modulus field - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2, RSA
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12k '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 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 MUST 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

	eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.38 Test case EAC2_ISO7816_J_38

Test - ID	EAC2_ISO7816_J_38
Purpose	Test a DV certificate while the Public Key DO has no public exponent field - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2, RSA
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12l '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 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.39 Test case EAC2_ISO7816_J_39

Test - ID	EAC2_ISO7816_J_39
Purpose	Test a DV certificate while the Public Key DO contains an unknown DO - Current date not updated
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the 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 MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 12" chapter as DV_CERT_12m '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 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 MUST 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 eID Card 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 MUST be used as returned by the PACE mechanism. 4. 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> • This certificate expiration date is BEFORE the effective date of the DV-Certificate used in step 2.
Expected results	1. '90 00' within a valid SM response 2. Checking error or warning processing '63 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response

3.5.40 Test case EAC2_ISO7816_J_40

Test - ID	EAC2_ISO7816_J_40
Purpose	Test the transition CVCA ⇨ IS key
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism.

	2. Send the appropriate IS-Certificate as specified in the "Certificate Set 10" chapter as IS_CERT_10. '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' within a valid SM response 2. Checking error or status bytes '63 00' within a valid SM response

3.5.41 Test case EAC2_ISO7816_J_41

Test - ID	EAC2_ISO7816_J_41
Purpose	Test the transition CVCA ⇨ domestic DV ⇨ CVCA
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 10" chapter as DV_CERT_10a. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate CA-Certificate as specified in the "Certificate Set 10" chapter as LINK_CERT_10. '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’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. Checking error or status bytes ’63 00’ within a valid SM response.

3.5.42 Test case EAC2_ISO7816_J_42

Test - ID	EAC2_ISO7816_J_42
Purpose	Test the transition CVCA ⇨ foreign DV ⇨ CVCA
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 10” chapter as DV_CERT_10b. ‘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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate CA-Certificate as specified in the “Certificate Set 10” chapter as LINK_CERT_10. ‘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’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. Checking error or status bytes ’63 00’ within a valid SM response.

3.5.43 Test case EAC2_ISO7816_J_43

Test - ID	EAC2_ISO7816_J_43
Purpose	Test the transition CVCA ⇨ domestic DV ⇨ domestic DV
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 10” chapter as DV_CERT_10a. ‘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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified in the “Certificate Set 10” chapter as DV_CERT_10c. ‘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’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response

	4. Checking error or status bytes ’63 00’ within a valid SM response.

3.5.44 Test case EAC2_ISO7816_J_44

Test - ID	EAC2_ISO7816_J_44
Purpose	Test the transition CVCA ⇨ domestic DV ⇨ foreign DV
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 10” chapter as DV_CERT_10a. ‘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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified in the “Certificate Set 10” chapter as DV_CERT_10d. ‘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’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. Checking error or status bytes ’63 00’ within a valid SM response.

3.5.45 Test case EAC2_ISO7816_J_45

Test - ID	EAC2_ISO7816_J_45
Purpose	Test the transition CVCA ⇨ foreign DV ⇨ domestic DV
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 10" chapter as DV_CERT_10b. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified in the "Certificate Set 10" chapter as DV_CERT_10c. '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' within a valid SM response. 2. '90 00' within a valid SM response. 3. '90 00' within a valid SM response. 4. Checking error or status bytes '63 00' within a valid SM response.

3.5.46 Test case EAC2_ISO7816_J_46

Test - ID	EAC2_ISO7816_J_46

Purpose	Test the transition CVCA ⇨ foreign DV ⇨ foreign DV
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 10" chapter as DV_CERT_10b. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate DV-Certificate as specified in the "Certificate Set 10" chapter as DV_CERT_10d. '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' within a valid SM response. 2. '90 00' within a valid SM response. 3. '90 00' within a valid SM response. 4. Checking error or status bytes '63 00' within a valid SM response.

3.5.47 Test case EAC2_ISO7816_J_47

Test - ID	EAC2_ISO7816_J_47
Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ foreign DV
Version	EAC2_1.0
Profile	PACE, TA2

Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 11" chapter as DV_CERT_11a. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 11" chapter as IS_CERT_11a. '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 DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the appropriate DV-Certificate as specified in the "Certificate Set 11" chapter as DV_CERT_11b. '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' within a valid SM response

	2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response 5. ’90 00’ or checking error within 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. Checking error or ’63 00’ within a valid SM response.

3.5.48 Test case EAC2_ISO7816_J_48

Test - ID	EAC2_ISO7816_J_48
Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ domestic DV
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 11” chapter as DV_CERT_11a. ‘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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 11” chapter as IS_CERT_11a. ‘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 DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the appropriate DV-Certificate as specified in the "Certificate Set 11" chapter as DV_CERT_11c. '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' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' or checking error within 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. Checking error or '63 00' within a valid SM response.

3.5.49 Test case EAC2_ISO7816_J_49

Test - ID	EAC2_ISO7816_J_49
Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ IS
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 11" chapter as DV_CERT_11a. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 11" chapter as IS_CERT_11a. '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 DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the appropriate IS-Certificate as specified in the "Certificate Set 11" chapter as IS_CERT_11b. '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' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' or checking error within 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. Checking error or '63 00' within a valid SM response.

3.5.50 Test case EAC2_ISO7816_J_50

Test - ID	EAC2_ISO7816_J_50
Purpose	Test the transition CVCA ⇨ DV ⇨ IS ⇨ CVCA
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed.

	2. All APDUs are sent as valid SecureMessaging APDUs. 3. All response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 11" chapter as DV_CERT_11a. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 11" chapter as IS_CERT_11a. '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 DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the appropriate CVCA-Certificate as specified in the "Certificate Set 11" chapter as LINK_CERT_11a. '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' within a valid SM response 2. '90 00' within a valid SM response

	3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response 5. ’90 00’ or checking error within 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. Checking error or ’63 00’ within a valid SM response.

3.5.51 Test case EAC2_ISO7816_J_51

Test - ID	EAC2_ISO7816_J_51
Purpose	Test a DV certificate with a wrong Public Key (shorter key length).
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism must have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference must be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 14” chapter as DV_CERT_14b. ‘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 key length of this certificate is different to the CVCA public key. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference given in the previous DVCA-Certificate sent. 4. Send the appropriate IS-Certificate as specified in the “Certificate Set 14” chapter as IS_CERT_14a. ‘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’ within a valid SM response 2. Checking error or ’63 00’ within a valid SM response. 3. ’90 00’ or checking error within 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. 4. Checking error or ’63 00’ within a valid SM response. Since the DV certificate was not verified successfully, it MUST NOT be possible to use it as the trust point for the IS-Certificate verification.

3.5.52 Test case EAC2_ISO7816_J_52

Test - ID	EAC2_ISO7816_J_52
Purpose	Test a IS certificate with a wrong Public Key (shorter key length).
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism must have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference must be used as returned by the PACE mechanism. 2. Send the appropriate CA-Certificate as specified in the “Certificate Set 14” chapter as DV_CERT_14aDV_CERT_14a. ‘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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference given in the previous DVCA-Certificate sent. 4. Send the appropriate IS-Certificate as specified in the “Certificate Set 14” chapter as IS_CERT_14b. ‘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 key length of this certificate is different to the CVCA and DV

	certificates public keys.
Expected results	1. ’90 00’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. Checking error or ’63 00’ within a valid SM response

3.6 Unit EAC2_ISO7816_K Terminal Authentication

This unit tests the second part of the terminal authentication process. In this step, the terminal proves the possession of the private key which belongs to its certificate.

PACE mechanism is performed with CAN (IS and ST) or PIN (AT). Used Certificate Holder Authorization Template MUST match terminal type and authorization given by the certificate chain.

3.6.1 Test case EAC2_ISO7816_K_1

Test - ID	EAC2_ISO7816_K_1
Purpose	Positive test with a valid terminal authentication process
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference> 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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.
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' within a valid SM response 6. '<Eight bytes of random data> 90 00' within a valid SM response 7. '90 00' within a valid SM response

3.6.2 Test case EAC2_ISO7816_K_2

Test - ID	EAC2_ISO7816_K_2
Purpose	Test with an invalid certificate reference for the MSE:Set AT command
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • To generate an invalid certification holder reference, the last character of the holder reference stored inside the IS-Certificate sent in step 4 is changed. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. If no error occurred yet, send the given external authenticate command to the eID Card. ‘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.
Expected results	1. ’90 00’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response 5. ’90 00’ or checking error within 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’ within a valid SM response or Checking error 7. Checking error or ‘6300’ within a valid SM response

3.6.3 Test case EAC2_ISO7816_K_3

Test - ID	EAC2_ISO7816_K_3
Purpose	Test with a terminal authentication process without secure messaging
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The APDU is step 1 - 6 are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card.

	‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘00 82 00 00 <Lc> <Terminal generated signature>’ • 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’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response 5. ’90 00’ within a valid SM response 6. ‘<Eight bytes of random data> 90 00’ within a valid SM response 7. Checking error as a plain response (without Secure Messaging)

3.6.4 Test case EAC2_ISO7816_K_4

Test - ID	EAC2_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	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used.

	4. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 5. Send the given external authenticate command to the eID Card. '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 DV_KEY_01.
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' or Checking error within a valid SM response 4. '<Eight bytes of random data> 90 00' or Checking error within a valid SM response 5. Checking error or '6300' within a valid SM response

3.6.5 Test case EAC2_ISO7816_K_5

Test - ID	EAC2_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	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certification Authority Reference as returned by the PACE mechanism. 2. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 3. Send the given external authenticate command to the eID Card. '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 CVCA_KEY_00.
Expected results	1. '90 00' or Checking error within a valid SM response 2. '<Eight bytes of random data> 90 00' or checking error within a valid SM response 3. Checking error or '6300' within a valid SM response

3.6.6 Test case EAC2_ISO7816_K_6

Test - ID	EAC2_ISO7816_K_6
Purpose	Test the external authenticate command with an invalid class byte
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All commands are encoded as legally structured Secure Messaging APDUs..
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used.

	6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '8C 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_01. • The class byte is set to an invalid value ('8C') 8. If the error code in step 7 was returned in a Secure Messaging response, verify that the secure messaging session has not been aborted. If a plain error code was returned, this step is skipped. Send an arbitrary SM APDU to the chip. '0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00'
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' within a valid SM response 6. '<Eight bytes of random data> 90 00' within a valid SM response 7. Checking error. Note that the behavior of the chip regarding the Secure Messaging context is undefined. Therefore this error can be returned in plain or as an SM response. 8. Skipped or '90 00' within a valid SM response

3.6.7 Test case EAC2_ISO7816_K_7

Test - ID	EAC2_ISO7816_K_7
Purpose	Terminal authentication process with two Get Challenge commands (Using the first challenge)
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given a second Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 8. Send the given external authenticate command to the eID Card. '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 signature is based on the first challenge received in step 6.
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' within a valid SM response 6. '<Eight bytes of random data> 90 00' within a valid SM response 7. '<Eight bytes of random data> 90 00' or Checking error within a valid SM response 8. Checking error or '63 00' within a valid SM response

3.6.8 Test case EAC2_ISO7816_K_8

Test - ID	EAC2_ISO7816_K_8
Purpose	Terminal authentication process with short challenge
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card.

	‘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 eID Card, otherwise skip this step. ‘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 signature is based on the short challenge received in step 6.
Expected results	1. ’90 00’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response 5. ’90 00’ within a valid SM response 6. ‘<Seven bytes of random data> 90 00’ within a valid SM response or Checking error 7. Skipped, Checking error or warning processing ’63 00’ within a valid SM response

3.6.9 Test case EAC2_ISO7816_K_9

Test - ID	EAC2_ISO7816_K_9
Purpose	Check the Terminal authentication – No Get Challenge Performed
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference>

	• The Certificate 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 IS_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given external authenticate command to the eID Card. '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_01. • The wrong signature is calculated without any challenge. 7. Perform CA2
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' within a valid SM response 6. Checking error or warning processing '63 00' within a valid SM response. 7. Checking error within a valid SM response during CA2

3.6.10 Test case EAC2_ISO7816_K_10

Test - ID	EAC2_ISO7816_K_10
Purpose	Check the Terminal authentication – No authentication key selection performed
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects

	83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 6. Send the given external authenticate command to the eID Card. '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_01. • The signature is based on the challenge received in step 5. 7. Perform CA2
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '<Eight bytes of random data> 90 00' or checking error within an SM response 6. Checking error or warning processing '63 00' within a valid SM response 7. Checking error within a valid SM response during CA2

3.6.11 Test case EAC2_ISO7816_K_11

Test - ID	EAC2_ISO7816_K_11

Purpose	Check the Terminal authentication – Wrong structure in the MSE: Set AT command
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 84 <L84> <Certificate Holder Reference > instead of tag 83 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'

	7. Send the given external authenticate command to the eID Card. '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_01. • The signature is based on the challenge received in step 6. 8. Perform CA2
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. Checking error within a valid SM response 6. '<Eight bytes of random data> 90 00' or checking error within an SM response 7. Checking error or warning processing '63 00' within a valid SM response 8. Checking error within a valid SM response during CA2

3.6.12 Test case EAC2_ISO7816_K_12

Test - ID	EAC2_ISO7816_K_12
Purpose	Check the Terminal authentication – Reset of the access rights in case of Application reset
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference>

	• The Certificate 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 IS_CERT_1. '0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_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 PACE mechanism • Perform CA2
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' within a valid SM response 6. '<Eight bytes of random data> 90 00' within an SM response 7. '90 00' within a valid SM response 8. Checking error within a valid SM response during CA2

3.6.13 Test case EAC2_ISO7816_K_13

Test - ID	EAC2_ISO7816_K_13
Purpose	This test case checks if the chip does not accept more than one execution of Terminal Authentication within the same session, same certificate set.
Version	EAC2_1.0
Profile	PACE, TA2

Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_1, IS_CERT_1). 3. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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> 00' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_CERT_1. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <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 eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature

Expected results	1. ’90 00’ within a valid SM response 2. ’90 00’ within a valid SM response 3. ’90 00’ within a valid SM response 4. ’90 00’ within a valid SM response 5. ’90 00’ or Checking error within a valid SM response. If this step returns Checking error the following steps don’t need to be performed. 6. ‘<Eight bytes of random data> 90 00’ within an SM response 7. Checking error within a valid SM response

3.6.14 Test case EAC2_ISO7816_K_14

Test - ID	EAC2_ISO7816_K_14
Purpose	This test case checks if the chip does not accept more than one execution of Terminal Authentication within the same session, different certificate sets.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_3, IS_CERT_3a). 3. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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> 00’ • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_CERT_1. ‘0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature
Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' or Checking error within a valid SM response. If this step returns Checking error the following steps don't need to be performed. 6. '<Eight bytes of random data> 90 00' within an SM response 7. Checking error within a valid SM response

3.6.15 Test case EAC2_ISO7816_K_15

Test - ID	EAC2_ISO7816_K_15
Purpose	This test case checks if the chip does not accept more than one execution of Terminal Authentication within the same session, different auxiliary data.
Version	EAC2_1.0
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17f). 3. Auxiliary data with valid Date of Birth data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Date of birth MUST NOT fit the required age. 4. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	• <Cryptogram> contains the following encrypted data objects
	`83 <L83> <Certification Authority Reference>`
	• The Certification Authority Reference MUST be used as returned by the PACE mechanism.
	2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17.
	`'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the following encrypted data objects
	`7F 4E <L7F4E> <certificate body>`
	`5F 37 <L5F37> <certificate signature>`
	3. Send the given MSE: Set DST APDU to the eID Card.
	`'0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the following encrypted data objects
	`83 <L83> <Certification Authority Reference>`
	• The Certificate 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 17" chapter as AT_CERT_17f.
	`'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the following encrypted data objects
	`7F 4E <L7F4E> <certificate body>`
	`5F 37 <L5F37> <certificate signature>`
	5. Send the given MSE: Set AT APDU to the eID Card.
	`'0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the following encrypted data objects
	`80 <L80> <Cryptographic Mechanism Reference>`
	`83 <L83> <Certificate Holder Reference>`
	`91 <L91> <Compressed Ephemeral Public Key>`
	`67 <L67> <Auxiliary Data>`
	• The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used.
	• Auxiliary data with valid Date of Birth data object MUST fit the required age.
	6. Send the given Get Challenge APDU to the eID Card.
	`'0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'`
	7. Send the given external authenticate command to the eID Card.
	`'0C 82 00 00 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the encrypted terminal generated signature

Expected results	1. '90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response

	4. ’90 00’ within a valid SM response 5. ’90 00’ or Checking error within a valid SM response. If this step returns Checking error the following steps don’t need to be performed. 6. ‘<Eight bytes of random data> 90 00’ within an SM response 7. Checking error within a valid SM response

3.6.16 Test case EAC2_ISO7816_K_16

Test - ID	EAC2_ISO7816_K_16
Purpose	Positive test with a valid terminal authentication process, but different order of commands (Get Challenge performed as first step in protocol)
Version	EAC2_1.1
Profile	PACE, TA2
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 2. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 3. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 4. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 5. Send the appropriate IS-Certificate as specified in the “Certificate Set 1” chapter as IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature>

	6. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference> 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' where the challenge of step 1 is used. • <Cryptogram> contains the encrypted terminal generated signature created with the private key of IS_KEY_01.
Expected results	1. '<Eight bytes of random data> 90 00' within a valid SM response 2. '90 00' within a valid SM response 3. '90 00' within a valid SM response 4. '90 00' within a valid SM response 5. '90 00' within a valid SM response 6. '90 00' within a valid SM response 7. '90 00' within a valid SM response

3.7 Unit EAC2_ISO7816_L Effective Access Conditions

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 terminal certificate.

All tests described here use following OIDs and DDOs within the PACE mechanism (tag '7F 4C'):

Profile	OID (terminal type)	DDO (relative authorization)
ePassport	id-IS (Inspection System)	23
eID	id-AT (Authentication Terminal)	3E 1F FF FF F7
eSign	id-ST (Signature Terminal)	03

These CHATs do not restrict access to any functionality.

Because eSign functionality is specified separately, the special functions "Install Qualified Certificate" and "Install Advanced Certificate" are not tested here.

3.7.1 Test case EAC2_ISO7816_L_1

Test - ID	EAC2_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	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 3" chapter as DV_CERT_3. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 3" chapter as IS_CERT_3a. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants only access to data group 3. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'

	7. Send the given external authenticate command to the eID Card. '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_03. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group 3 content data> 90 00' within a valid Secure Messaging response.

3.7.2 Test case EAC2_ISO7816_L_2

Test - ID	EAC2_ISO7816_L_2
Purpose	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	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 3" chapter as DV_CERT_3.

	‘0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 3” chapter as IS_CERT_3a. ‘0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants only access to data group 3. 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘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_03. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. true 9. ’90 00’ within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.7.3 Test case EAC2_ISO7816_L_3

Test - ID	EAC2_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	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 3” chapter as DV_CERT_3 ‘0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 3" chapter as IS_CERT _3b. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants only access to data group 4. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_03. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group 4 content data> 90 00' within a valid Secure Messaging response.

3.7.4 Test case EAC2_ISO7816_L_4

Test - ID	EAC2_ISO7816_L_4
Purpose	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	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 3" chapter as DV_CERT_3 '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 3 and 4. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 3" chapter as IS_CERT _3b. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants only access to data group 4. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key>

	• The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_03. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. Checking error

3.7.5 Test case EAC2_ISO7816_L_5

Test - ID	EAC2_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	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference>

	• The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 4" chapter as DV_CERT_4 '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 3 only. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 4" chapter as IS_CERT_4. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_04. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID.

	10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group 3 content data> 90 00' within a valid Secure Messaging response.

3.7.6 Test case EAC2_ISO7816_L_6

Test - ID	EAC2_ISO7816_L_6
Purpose	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	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 4" chapter as DV_CERT_4 '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 3 only. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference>

	• The Certificate 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 4" chapter as IS_CERT_4. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_04. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.7.7 Test case EAC2_ISO7816_L_7

Test - ID	EAC2_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 enables access to both data 3 and 4.
Version	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 5" chapter as DV_CERT_5 '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants access to data group 4 only. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 5" chapter as IS_CERT_5. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This IS-Certificate grants access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference >

	91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_05. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group 4 content data> 90 00' within a valid Secure Messaging response.

3.7.8 Test case EAC2_ISO7816_L_8

Test - ID	EAC2_ISO7816_L_8
Purpose	Test that data group 3 cannot be accessed if the DV certificate grants access to data group 4 only and the IS certificate enables access to both data group 3 and 4.
Version	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

- <Cryptogram> contains the following encrypted data objects
  `83 <L83> <Certification Authority Reference>`
- The Certification Authority Reference MUST be used as returned by the PACE mechanism.

2. Send the appropriate DV-Certificate as specified in the "Certificate Set 5" chapter as 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 <L7F4E> <certificate body>`
     `5F 37 <L5F37> <certificate signature>`
   - This DV-Certificate grants access to data group 4 only.
3. Send the given MSE: Set DST APDU to the eID Card.
   `'0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
   - <Cryptogram> contains the following encrypted data objects
     `83 <L83> <Certification Authority Reference>`
   - The Certificate 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 5" chapter as IS_CERT_5.
   `'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>'`
   - <Cryptogram> contains the following encrypted data objects
     `7F 4E <L7F4E> <certificate body>`
     `5F 37 <L5F37> <certificate signature>`
   - This IS-Certificate grants access to data group 3 and 4.
5. Send the given MSE: Set AT APDU to the eID Card.
   `'0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
   - <Cryptogram> contains the following encrypted data objects
     `80 <L80> <Cryptographic Mechanism Reference>`
     `83 <L83> <Certificate Holder Reference >`
     `91 <L91> <Compressed Ephemeral Public Key>`
   - The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used.
6. Send the given Get Challenge APDU to the eID Card.
   `'0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'`
7. Send the given external authenticate command to the eID Card.
   `'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_05.
8. The Chip Authentication mechanism MUST be performed.
9. Send the given Select Application APDU to the eID Card (selecting ePassport application):
   `'0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08`

	<Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.7.9 Test case EAC2_ISO7816_L_9

Test - ID	EAC2_ISO7816_L_9
Purpose	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	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference>

	• The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <L_C> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. The Chip Authentication mechanism MUST be performed. 8. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 9. If the previous step returned an error, skip this step. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. false 8. '90 00' or checking error within a valid Secure Messaging response. 9. Skipped or checking error within a valid Secure Messaging response.

3.7.10 Test case EAC2_ISO7816_L_10

Test - ID	EAC2_ISO7816_L_10
Purpose	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	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <L_C> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. The Chip Authentication mechanism MUST be performed. 8. Send the given Select Application APDU to the eID Card (selecting ePassport application):

	‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted ePassport application-ID. 9. If the previous step returned an error, skip this step. Send the given Read Binary (with SFI) command to the eID Card, 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’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. false 8. ’90 00’ or checking error within a valid Secure Messaging response. 9. Skipped or checking error within a valid Secure Messaging response.

3.7.11 Test case EAC2_ISO7816_L_11

Test - ID	EAC2_ISO7816_L_11
Purpose	Test with a failed external authenticate command does not enable the access to the sensitive data in data group 3.
Version	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects

	83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. If the previous step returned an error, skip this step. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. Checking error or warning processing '63 00' within a valid Secure Messaging response.

	8. false 9. ’90 00’ or checking error within a valid Secure Messaging response. 10. Skipped or checking error within a valid Secure Messaging response

3.7.12 Test case EAC2_ISO7816_L_12

Test - ID	EAC2_ISO7816_L_12
Purpose	Test with a failed external authenticate command does not enable the access to the sensitive data in data group 4.
Version	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate 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 IS_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 <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference>

	83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. If the previous step returned an error, skip this step. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. Checking error or warning processing '63 00' within a valid Secure Messaging response. 8. false 9. '90 00' or checking error within a valid Secure Messaging response. 10. Skipped or checking error within a valid Secure Messaging response.

3.7.13 Test case EAC2_ISO7816_L_13 Template

Test - ID	EAC2_ISO7816_L_13_template
Purpose	Positive test with a valid terminal authentication process with read access permission for well defined DGs if the DV certificate permits read access to all DGs while the Terminal certificate restricts access to one DG. DV certificate is an official domestic certificate.
Version	See Table 10
Profile	eID, TA2, required data group presence see Table 10
Preconditions	1. The PACE mechanism MUST have been performed (CAN).

	2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 19" chapter as DV_CERT_19. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants read access to all data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 19" chapter as defined in Table 10, column Cert Reference '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This Terminal-Certificate grants access to data groups as defined in Table 10, column Access Rules. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference> 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature

	8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, to verify the read access to the selected data group has been granted. '0C B0 (80 | <SFI>) 00 0D 97 01 01 8E 08 <Checksum> 00' • <SFI> contains the SFI reference as defined in Table 10, column *SFI*.
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group content data> 90 00' within a valid Secure Messaging response.

Test case EAC2_ISO7816_L_13a to Test case EAC2_ISO7816_L_13v:

Test Case ID	Version	Access Rules	Cert Reference	SFI
EAC2_ISO7816_L_13a	EAC2_1.0	This terminal certificate grants only read access to data group 1	AT_CERT_19a	0x01
EAC2_ISO7816_L_13b	EAC2_1.0	This terminal certificate grants only read access to data group 2	AT_CERT_19b	0x02
EAC2_ISO7816_L_13c	EAC2_1.0	This terminal certificate grants only read access to data group 3	AT_CERT_19c	0x03
EAC2_ISO7816_L_13d	EAC2_1.0	This terminal certificate grants only read access to data group 4	AT_CERT_19d	0x04
EAC2_ISO7816_L_13e	EAC2_1.0	This terminal certificate grants only read access to data group 5	AT_CERT_19e	0x05
EAC2_ISO7816_L_13f	EAC2_1.0	This terminal certificate grants only read access to data group 6	AT_CERT_19f	0x06
EAC2_ISO7816_L_13g	EAC2_1.0	This terminal certificate grants only read access to data group 7	AT_CERT_19g	0x07
EAC2_ISO7816_L_13h	EAC2_1.0	This terminal certificate grants only read access to data group 8	AT_CERT_19h	0x08
EAC2_ISO7816_L_13i	EAC2_1.0	This terminal certificate grants only read access to data group 9	AT_CERT_19i	0x09
EAC2_ISO7816_L_13j	EAC2_1.0	This terminal certificate grants only read access to data group 10	AT_CERT_19j	0x0a
EAC2_ISO7816_L_13k	EAC2_1.0	This terminal certificate grants only read access to data group 11	AT_CERT_19k	0x0b
EAC2_ISO7816_L_13l	EAC2_1.0	This terminal certificate grants only read access to data group 12	AT_CERT_19l	0x0c
EAC2_ISO7816_L_13m	EAC2_1.0	This terminal certificate grants only read access to data group 13	AT_CERT_19m	0x0d
EAC2_ISO7816_L_13n	EAC2_1.0	This terminal certificate grants only read access to data group 14	AT_CERT_19n	0x0e
EAC2_ISO7816_L_13o	EAC2_1.0	This terminal certificate grants only read access to data group 15	AT_CERT_19o	0x0f
EAC2_ISO7816_L_13p	EAC2_1.0	This terminal certificate grants only read access to data group 16	AT_CERT_19p	0x10
EAC2_ISO7816_L_13q	EAC2_1.0	This terminal certificate grants only read access to data group 17	AT_CERT_19q	0x11
EAC2_ISO7816_L_13r	EAC2_1.0	This terminal certificate grants only read access to data group 18	AT_CERT_19r	0x12
EAC2_ISO7816_L_13s	EAC2_1.0	This terminal certificate grants only read access to data group 19	AT_CERT_19s	0x13
EAC2_ISO7816_L_13t	EAC2_1.0	This terminal certificate grants only read access to data group 20	AT_CERT_19t	0x14
EAC2_ISO7816_L_13u	EAC2_1.0	This terminal certificate grants only read access to data group 21	AT_CERT_19u	0x15
EAC2_ISO7816_L_13v	EAC2_1.1	This terminal certificate grants only read access to data group 22	AT_CERT_19v	0x16

Table 10: Test cases EAC2_ISO7816_L_13

3.7.14 Test case EAC2_ISO7816_L_14 Template

Test - ID	EAC2_ISO7816_L_14_template
Purpose	Positive test with a valid terminal authentication process with read access permission for well defined DGs if the DV certificate permits read access to all DGs while the Terminal certificate restricts access to one DG. DV certificate is a non-official certificate.
Version	See Table 11
Profile	eID, TA2, required data group presence see Table 11
Preconditions	1. The PACE mechanism MUST have been performed (PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 20" chapter as DV_CERT_20. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants read access to all data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 20" chapter as defined in Table 11, column Cert Reference '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This Terminal-Certificate grants access to data groups as defined in Table 11, column Access Rules 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	• <Cryptogram> contains the following encrypted data objects
	`80 <L80> <Cryptographic Mechanism Reference>`
	`83 <L83> <Certificate Holder Reference >`
	`91 <L91> <Compressed Ephemeral Public Key>`
	• The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used.
	6. Send the given Get Challenge APDU to the eID Card.
	`'0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'`
	7. Send the given external authenticate command to the eID Card.
	`'0C 82 00 00 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the encrypted terminal generated signature
	8. The Chip Authentication mechanism MUST be performed.
	9. Send the given Select Application APDU to the eID Card (selecting eID application):
	`'0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
	• <Cryptogram> contains the encrypted eID application-ID.
	10. Send the given Read Binary (with SFI) command to the eID Card, to verify the read access to the selected data group has been granted.
	`'0C B0 (80 | <SFI>) 00 0D 97 01 01 8E 08 <Checksum> 00'`
	• <SFI> contains the SFI reference as defined in Table 11, column *SFI*.

Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group content data> 90 00' within a valid Secure Messaging response.

Test case EAC2_ISO7816_L_14a to Test case EAC2_ISO7816_L_14v

Test Case ID	Version	Access Rules	Cert Reference	SFI
EAC2_ISO7816_L_14a	EAC2_1.0	This terminal certificate grants only read access to data group 1	AT_CERT_20a	0x01
EAC2_ISO7816_L_14b	EAC2_1.0	This terminal certificate grants only read access to data group 2	AT_CERT_20b	0x02
EAC2_ISO7816_L_14c	EAC2_1.0	This terminal certificate grants only read access to data group 3	AT_CERT_20c	0x03
EAC2_ISO7816_L_14d	EAC2_1.0	This terminal certificate grants only read access to data group 4	AT_CERT_20d	0x04
EAC2_ISO7816_L_14e	EAC2_1.0	This terminal certificate grants only read access to data group 5	AT_CERT_20e	0x05
EAC2_ISO7816_L_14f	EAC2_1.0	This terminal certificate grants only read access to data group 6	AT_CERT_20f	0x06
EAC2_ISO7816_L_14g	EAC2_1.0	This terminal certificate grants only read access to data group 7	AT_CERT_20g	0x07
EAC2_ISO7816_L_14h	EAC2_1.0	This terminal certificate grants only read access to data group 8	AT_CERT_20h	0x08
EAC2_ISO7816_L_14i	EAC2_1.0	This terminal certificate grants only read access to data group 9	AT_CERT_20i	0x09
EAC2_ISO7816_L_14j	EAC2_1.0	This terminal certificate grants only read access to data group 10	AT_CERT_20j	0x0a
EAC2_ISO7816_L_14k	EAC2_1.0	This terminal certificate grants only read access to data group 11	AT_CERT_20k	0x0b
EAC2_ISO7816_L_14l	EAC2_1.0	This terminal certificate grants only read access to data group 12	AT_CERT_20l	0x0c
EAC2_ISO7816_L_14m	EAC2_1.0	This terminal certificate grants only read access to data group 13	AT_CERT_20m	0x0d
EAC2_ISO7816_L_14n	EAC2_1.0	This terminal certificate grants only read access to data group 14	AT_CERT_20n	0x0e
EAC2_ISO7816_L_14o	EAC2_1.0	This terminal certificate grants only read access to data group 15	AT_CERT_20o	0x0f
EAC2_ISO7816_L_14p	EAC2_1.0	This terminal certificate grants only read access to data group 16	AT_CERT_20p	0x10
EAC2_ISO7816_L_14q	EAC2_1.0	This terminal certificate grants only read access to data group 17	AT_CERT_20q	0x11
EAC2_ISO7816_L_14r	EAC2_1.0	This terminal certificate grants only read access to data group 18	AT_CERT_20r	0x12
EAC2_ISO7816_L_14s	EAC2_1.0	This terminal certificate grants only read access to data group 19	AT_CERT_20s	0x13
EAC2_ISO7816_L_14t	EAC2_1.0	This terminal certificate grants only read access to data group 20	AT_CERT_20t	0x14
EAC2_ISO7816_L_14u	EAC2_1.0	This terminal certificate grants only read access to data group 21	AT_CERT_20u	0x15
EAC2_ISO7816_L_14v	EAC2_1.1	This terminal certificate grants only read access to data group 22	AT_CERT_20v	0x16

Table 11: Test cases EAC2_ISO7816_L_14

3.7.15 Test case EAC2_ISO7816_L_15 Template

Test - ID	EAC2_ISO7816_L_15_template
Purpose	Positive test with a valid terminal authentication process with write access permission for well defined DGs if the DV certificate permits write access to all writable DGs while the Terminal certificate restricts access to one DG. DV certificate is an official domestic certificate
Version	See Table 12
Profile	eID, TA2, required data group presence see Table 12
Preconditions	1. The PACE mechanism MUST have been performed (CAN). 2. All APDUs are sent as valid SecureMessaging APDUs. 3. Read content of DG 17 to DG 22 to restore the content after this test scenario using DV_CERT_19 and AT_CERT_19q to AT_CERT_19v
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 21" chapter as DV_CERT_21. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This DV-Certificate grants write access to all writable data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 21" chapter as referenced in Table 12, column Cert Reference '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L7F4E> <certificate body> 5F 37 <L5F37> <certificate signature> • This Terminal-Certificate grants access to data groups as defined in Table 12, column Access Rules. 5. Send the given MSE: Set AT APDU to the eID Card.

	‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L80> <Cryptographic Mechanism Reference> 83 <L83> <Certificate Holder Reference > 91 <L91> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Update Binary (with SFI) command to the eID Card, to verify that write access to the selected data group has been granted. ‘0C D6 (80 | <SFI>) 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: 01 02 03 04 • <SFI> contains the SFI reference as defined in Table 12, column SFI.
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ‘90 00’ within a valid Secure Messaging response.
Post processing	1. Restore original content of DG 17 to DG 22

Test case EAC2_ISO7816_L_15a to Test case EAC2_ISO7815_L_15f:

Test Case ID	Version	Access Rules	Cert Reference	SFI
EAC2_ISO7816_L_15a	EAC2_1.03	This terminal certificate grants only r/w access to data group 17	AT_CERT_21a	0x11
EAC2_ISO7816_L_15b	EAC2_1.0	This terminal certificate grants	AT_CERT_21b	0x12

	3	only r/w access to data group 18		
EAC2_ISO7816_L_15c	EAC2_1.03	This terminal certificate grants only r/w access to data group 19	AT_CERT_21c	0x13
EAC2_ISO7816_L_15d	EAC2_1.03	This terminal certificate grants only r/w access to data group 20	AT_CERT_21d	0x14
EAC2_ISO7816_L_15e	EAC2_1.03	This terminal certificate grants only r/w access to data group 21	AT_CERT_21e	0x15
EAC2_ISO7816_L_15f	EAC2_1.1	This terminal certificate grants only r/w access to data group 22	AT_CERT_21f	0x16

Table 12: Test cases EAC2_ISO7816_L_15

3.7.16 Test case EAC2_ISO7816_L_16 Template

Test - ID	EAC2_ISO7816_L_16_template
Purpose	Positive test with a valid terminal authentication process with write access permission for well defined DGs if the DV certificate permits write access to all writable DGs while the Terminal certificate restricts access to one DG. DV certificate is a non-official certificate
Version	See Table 13
Profile	eID, TA2, required data group presence see Table 13
Preconditions	1. The PACE mechanism MUST have been performed (PIN). 2. All APDUs are sent as valid SecureMessaging APDUs. 3. Read content of DG 17 to DG 22 to restore the content after this test scenario using DV_CERT_19 and AT_CERT_19q to AT_CERT_19v
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 22" chapter as DV_CERT_22. '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 write access to all writable data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent

	in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 22" chapter as defined in Table 13, column Cert Reference. '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 Terminal-Certificate grants only write access to data groups as defined in Table 13, column Access Rules. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '00 A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 10. Send the given Update Binary (with SFI) command to the eID Card, to verify that write access to the selected data group has been granted. '0C D6 (80 | <SFI>) 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: 01 02 03 04 • <SFI> contains the SFI reference as defined in Table 13, column SFI
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response.

	8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ‘90 00’ within a valid Secure Messaging response.
Post processing	1. Restore original content of DG 17 to DG 22

Test case EAC2_ISO7816_L_16a to Test case EAC2_ISO7816_L_16e:

Test Case ID	Version	Access Rules	Cert Reference	SFI
EAC2_ISO7816_L_16a	EAC2_1.03	This terminal certificate grants only r/w access to data group 17	AT_CERT_22a	0x11
EAC2_ISO7816_L_16b	EAC2_1.03	This terminal certificate grants only r/w access to data group 18	AT_CERT_22b	0x12
EAC2_ISO7816_L_16c	EAC2_1.03	This terminal certificate grants only r/w access to data group 19	AT_CERT_22c	0x13
EAC2_ISO7816_L_16d	EAC2_1.03	This terminal certificate grants only r/w access to data group 20	AT_CERT_22d	0x14
EAC2_ISO7816_L_16e	EAC2_1.03	This terminal certificate grants only r/w access to data group 21	AT_CERT_22e	0x15
EAC2_ISO7816_L_16f	EAC2_1.1	This terminal certificate grants only r/w access to data group 22	AT_CERT_22f	0x16

Table 13: Test cases EAC2_ISO7816_L_16

3.7.17 Test case EAC2_ISO7816_L_17

Test - ID	EAC2_ISO7816_L_17
Purpose	Positive test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is Age Verification.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17f. '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 Terminal-Certificate grants access to special function "Age Verification" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> 67 <L_{67}> <Auxiliary Data> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. • Auxiliary Data contains valid Date of Birth data. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response.

	5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ‘90 00’ within a valid Secure Messaging response.

3.7.18 Test case EAC2_ISO7816_L_18

Test - ID	EAC2_ISO7816_L_18
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is “Install Qualified Certificate” but “Age Verification” is used.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 17” chapter as DV_CERT_17. ‘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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the “Certificate Set 17” chapter as AT_CERT_17d.

	‘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 Terminal-Certificate grants access to special function “Install Qualified Certificate” 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> 67 <L_{67}> <Auxiliary Data> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. • Auxiliary Data contains valid Date of Birth data. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response.

	10. ’69 82’ within a valid Secure Messaging response.

3.7.19 Test case EAC2_ISO7816_L_19

Test - ID	EAC2_ISO7816_L_19
Purpose	Positive test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is Municipality ID Verification.
Version	EAC2_1.03
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 17” chapter as DV_CERT_17. ‘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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the “Certificate Set 17” chapter as AT_CERT_17g. ‘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 Terminal-Certificate grants access to special function “Municipality ID Verification” 5. Send the given MSE: Set AT APDU to the eID Card.

	‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> 67 <L_{67}> <Auxiliary Data> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. • Auxiliary Data contains valid Municipality ID data. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ‘90 00’ within a valid Secure Messaging response.

3.7.20 Test case EAC2_ISO7816_L_20

Test - ID	EAC2_ISO7816_L_20
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate

	restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is “Install Qualified Certificate” but “Municipality ID Verification” is used.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 17” chapter as DV_CERT_17. ‘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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the “Certificate Set 17” chapter as AT_CERT_17d. ‘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 Terminal-Certificate grants access to special function “Install Qualified Certificate” 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> 67 <L_{67}> <Auxiliary Data>

	• The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. • Auxiliary Data contains valid Municipality ID data. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response. 10. '69 82'. The error MUST be encoded in a valid Secure Messaging response.

3.7.21 Test case EAC2_ISO7816_L_21

Test - ID	EAC2_ISO7816_L_21
Version	deleted in version 1.00 RC

3.7.22 Test case EAC2_ISO7816_L_22

Test - ID	EAC2_ISO7816_L_22
Version	deleted in version 1.00 RC

3.7.23 Test case EAC2_ISO7816_L_23

Test - ID	EAC2_ISO7816_L_23
Purpose	Positive test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is “CAN allowed”.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (using CAN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 17” chapter as DV_CERT_17. ‘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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the “Certificate Set 17” chapter as AT_CERT_17a. ‘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 Terminal-Certificate grants access to special function “CAN allowed” 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’

	• <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '0C A4 04 0C <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' 10. Send the given Read Binary (with SFI) command to the eID Card, to verify the access to the data group 1 has been granted. '0C B0 81 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response. 10. '90 00' within a valid Secure Messaging response.

3.7.24 Test case EAC2_ISO7816_L_24

Test - ID	EAC2_ISO7816_L_24
Version	deleted in version 1.00 RC

3.7.25 Test case EAC2_ISO7816_L_25

Test - ID	EAC2_ISO7816_L_25
Purpose	Positive test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is "PIN Management". Deactivate PIN within pin management is tested.

Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (using PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17b. '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 Terminal-Certificate grants access to special function "PIN Management" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'

	7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Deactivate PIN APDU to the eID Card: '0C 04 10 03 <Lc> 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response.

3.7.26 Test case EAC2_ISO7816_L_26

Test - ID	EAC2_ISO7816_L_26
Purpose	Positive test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is "PIN Management". Activate PIN within pin management is tested.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. PIN MUST have been deactivated (see Test case EAC2_ISO7816_L_25). 2. The PACE mechanism MUST have been performed (using CAN). 3. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17b. '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 Terminal-Certificate grants access to special function "PIN Management" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Activate PIN APDU to the eID Card: '0C 44 10 03 <Lc> 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response.

3.7.27 Test case EAC2_ISO7816_L_27

Test - ID	EAC2_ISO7816_L_27
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is “Install Qualified Certificate” but “PIN Management” is used.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (using PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 17” chapter as DV_CERT_17. ‘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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the “Certificate Set 17” chapter as AT_CERT_17d. ‘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 Terminal-Certificate grants access to special function “Install Qualified Certificate” 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’

	• <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Deactivate PIN APDU to the eID Card: '0C 04 10 03 <Lc> 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '69 82'. The error MUST be encoded in a valid Secure Messaging response.

3.7.28 Test case EAC2_ISO7816_L_28

Test - ID	EAC2_ISO7816_L_28
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is "Install Qualified Certificate" but "PIN Management" is used.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed (using PIN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by

	the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17d. '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 Terminal-Certificate grants access to special function "Install Qualified Certificate" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Deactivate PIN APDU to the eID Card: '0C 04 10 03 <Lc> 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response.

	4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’69 82’. The error MUST be encoded in a valid Secure Messaging response.

3.7.29 Test case EAC2_ISO7816_L_29

Deleted in version 1.1

3.7.30 Test case EAC2_ISO7816_L_30

Deleted in version 1.1

3.7.31 Test case EAC2_ISO7816_L_31

Deleted in version 1.1

3.7.32 Test case EAC2_ISO7816_L_32

Deleted in version 1.1

3.7.33 Test case EAC2_ISO7816_L_33

Deleted in version 1.1

3.7.34 Test case EAC2_ISO7816_L_34

Deleted in version 1.1

3.7.35 Test case EAC2_ISO7816_L_35

Test - ID	EAC2_ISO7816_L_35
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. It is tested that RFU bits are ignored.
Version	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE (MRZ) mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card.

	‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 3” chapter as DV_CERT_3a. ‘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 and has all RFU bits set to 1. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 3” chapter as IS_CERT_3c. ‘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 has all RFU bits set to 1. 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘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_03.

	8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. '<first byte of data group 3 content data> 90 00' within a valid Secure Messaging response.

3.7.36 Test case EAC2_ISO7816_L_36

Test - ID	EAC2_ISO7816_L_36
Purpose	Positive test with a valid terminal authentication process with read access permission for DG 1 if the DV certificate permits read access to all DGs while the terminal certificate restricts access to DG 1. DV certificate is an official domestic certificate. It is tested that RFU bits are ignored.
Version	EAC2_1.0
Profile	eID, TA2, DG1
Preconditions	1. The PACE mechanism MUST have been performed (CAN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 19" chapter as DV_CERT_19a. '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 read access to all data groups and has all RFU bits set to 1. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 19" chapter as AT_CERT_19w. '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 Terminal-Certificate grants access to data group 1 and has all RFU bits set to 1. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, to verify the read access to the selected data group has been granted. '0C B0 81 00 0D 97 01 01 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response.

	3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ‘<first byte of data group content data> 90 00’ within a valid Secure Messaging response.

3.7.37 Test case EAC2_ISO7816_L_37

Test - ID	EAC2_ISO7816_L_37
Purpose	Positive test with a valid terminal authentication process. The DV certificate permits all special functions while the terminal certificate restricts access to one special function. The DV certificate is an official domestic certificate. The special function allowed by terminal certificate is “Privileged Terminal”.
Version	EAC2_1.1
Profile	eID, TA2, CS
Preconditions	1. The PACE mechanism MUST have been performed (CAN). 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 17” chapter as DV_CERT_17. ‘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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the “Certificate

	Set 17” chapter as AT_CERT_17h. ‘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 Terminal-Certificate grants access to special function “Privileged Terminal” 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> 91 <L_{91}> <Compressed Ephemeral Public Key> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. Send the given Read Binary (with SFI) command to the eID Card, to verify the read access to the selected data group has been granted. ‘0C B0 (80 | <sfi.EF.ChipSecurity> 00 0D 97 01 01 8E 08 <Checksum> 00’
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. ‘<One byte content of EF.ChipSecurity> 90 00’ within a valid Secure Messaging response.

3.8 Unit EAC2_ISO7816_M Update mechanism

This unit contains all test cases, which update the chip’s 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. Also, this unit should be performed from first to last test case in the given order.

The following diagram shows the movement of the chip's current date (arrow at top) and the trust points (bars) for ePassport and eID (moved by link certificates). Note: Test cases M_6 and M_8 do not change the chip's persistent memory.

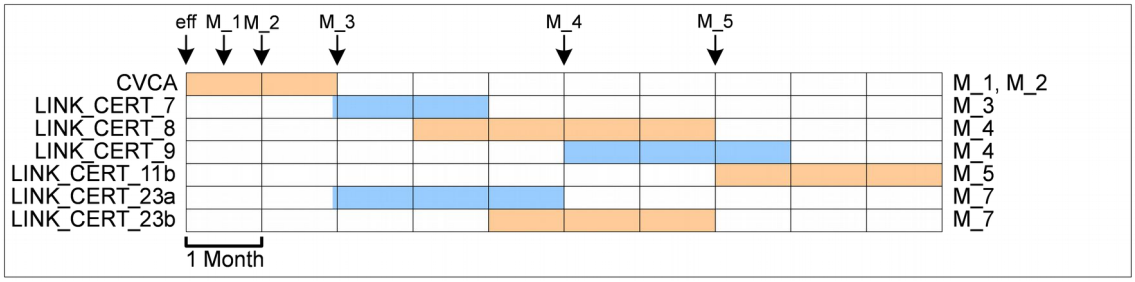

Figure 2: Test unit M overview

3.8.1 Test case EAC2_ISO7816_M_1

Test - ID	EAC2_ISO7816_M_1
Purpose	Test the "Current Date" update mechanism with a new domestic IS certificate. This test works with IS trust points.
Version	EAC2_1.0
Profile	TA2
Preconditions	1. The PACE mechanism MUST have been performed as IS using CAN. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 6" as DV_CERT_6 '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 DV certificate is marked as a domestic certificate 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 6"

	as IS_CERT_6a. '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 has an advanced effective date. Since the DV certificate was marked as a domestic one, the chip MUST 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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 6. Send the appropriate DV-Certificate as specified in the "Certificate Set 6" as DV_CERT_6 '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 DV certificate is marked as a domestic certificate 7. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 6 has to be used. 8. Send the appropriate IS-Certificate as specified in the "Certificate Set 6" as IS_CERT_6b. '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 has an expiry date BEFORE the effective of the IS certificate used in step 4. Therefore this certificate MUST be rejected.
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response.. 3. '90 00' within a valid Secure Messaging response.. 4. '90 00' within a valid Secure Messaging response..

	5. ’90 00’ within a valid Secure Messaging response.. 6. ’90 00’ within a valid Secure Messaging response.. 7. ’90 00’ within a valid Secure Messaging response.. 8. Checking error or ‘6300’ within a valid Secure Messaging response. This certificate MUST no longer be valid, since the current date of the chip has been updated.

3.8.2 Test case EAC2_ISO7816_M_2

Test - ID	EAC2_ISO7816_M_2
Purpose	Test the “Current Date” update mechanism with a new DV certificate. This test works with IS trust points.
Version	EAC2_1.0
Profile	TA2
Preconditions	1. The PACE mechanism MUST have been performed as IS using CAN. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 6” as DV_CERT_6a ‘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 DV certificate has an advanced effective date beyond the expiration date of DV_CERT_6 • Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 4. Send the appropriate DV-Certificate as specified in the “Certificate Set 6” as DV_CERT_6. ‘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 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. Checking error or '6300' within a valid Secure Messaging response. This certificate MUST no longer be valid, since the current date of the chip has been updated.

3.8.3 Test case EAC2_ISO7816_M_3

Test - ID	EAC2_ISO7816_M_3
Purpose	Test the "Trust Point" update mechanism with a new link certificate. This test changes the IS trust points.
Version	EAC2_1.0
Profile	TA2
Preconditions	1. The PACE mechanism MUST have been performed as IS using CAN. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate link certificate as specified in the "Certificate Set 7" as LINK_CERT_7. The ePassport MUST update the trust point with this new certificate. '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. 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 again and perform PACE again and verify that the new trust point is at the first position (i.e. DO87) and the previous one has been moved to the second position (i.e. DO88). 4. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects

	83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the trust point received in DO 88 as returned by the PACE mechanism. 5. Send the appropriate DV-Certificate as specified in the "Certificate Set 7" as DV_CERT_7a. '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> • Since the previous trust point is still valid, the certificate MUST 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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the trust point received in DO 87 as returned by the PACE mechanism. 7. Send the appropriate DV-Certificate as specified in the "Certificate Set 7" as DV_CERT_7b. '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> • Since the effective date of this certificate is after the expiration date of the original trust point, the chip MUST update the current date and MUST also disable the original trust point for DV certificate verification. • Reset the chip after this step and restore the preconditions for this test case before the next step is performed. 8. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • Use the original Certification Authority Reference (same as in step 4). 9. Send the appropriate DV-Certificate as specified in the "Certificate Set 7" as DV_CERT_7a. '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>

	• Since the trust point has been disabled for DV certificate verification, the certificate verification MUST fail.
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. true 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ’90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. ’90 00’ or checking error within a valid Secure Messaging response. 9. Checking error or ‘6300’ within a valid Secure Messaging response.. This certificate MUST no longer be valid, since the current date of the chip has been updated.

3.8.4 Test case EAC2_ISO7816_M_4

Before performing this test case, validate, that the trust point has successfully been updated in test case EAC2_ISO7816_M_3.

Test - ID	EAC2_ISO7816_M_4
Purpose	Test the “Trust Point” update mechanism with two link certificates. This test changes the IS trust points.
Version	EAC2_1.0
Profile	TA2
Preconditions	1. The PACE mechanism MUST have been performed as IS using CAN. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. This test case can only be done AFTER EAC2_ISO7816_M_3 has been performed.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the trust point received in DO 87 as read returned by the PACE mechanism. 2. Send the appropriate link certificate as specified in the “Certificate Set 8” as LINK_CERT_8. The ePassport MUST update the trust point with this new certificate. ‘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 eID Card. ‘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 reference> • The Certification Authority Reference MUST be used as specified in the Link certificate used in step 2. 4. Send the appropriate link certificate as specified in the "Certificate Set 9" as "LINK_CERT_9". The ePassport MUST update the trust point with this new certificate. '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 DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as specified in the second Link certificate used in step 4. 6. Send the appropriate DV-Certificate as specified in the "Certificate Set 9" as DV_CERT_9. '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> 7. Power off the chip, perform PACE again and verify the trust points returned by the PACE mechanism. Both new trust points must be present. The previous trust point from the LINK_CERT_7 MUST be gone.
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '90 00' within a valid Secure Messaging response. 7. true

3.8.5 Test case EAC2_ISO7816_M_5

Before performing this test case, validate, that the trust point has successfully been updated in test cases EAC2_ISO7816_M_3 and EAC2_ISO7816_M_4.

Test - ID	EAC2_ISO7816_M_5
Purpose	Test the transition CVCA ⇨ CVCA ⇨ IS. This test changes the IS trust points.

Version	EAC2_1.0
Profile	TA2
Preconditions	1. The PACE mechanism MUST have been performed as IS using CAN. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. This test case can only be done AFTER EAC2_ISO7816_M_4 has been performed.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism (Primary trust point, i.e. DO87). 2. Send the appropriate CA-Certificate as specified in the "Certificate Set 11" chapter as LINK_CERT_11b. '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 eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the new CVCA-Certificate sent in step 2 has to be used. 4. Send the appropriate IS-Certificate as specified in the "Certificate Set 11" chapter as IS_CERT_11c. '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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. Checking error or '63 00' in a SM response

3.8.6 Test case EAC2_ISO7816_M_6

Before performing this test case, validate, that the trust point has successfully been updated in test case EAC2_ISO7816_M_3.

Test - ID	EAC2_ISO7816_M_6

Purpose	Test the "Trust Point" update mechanism dependency to other applications, i. e. all applications share the same current date, but have different trust points. This test works with AT trust points.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. Validate that EAC2_ISO7816_M_3 has been performed successfully 2. The PACE mechanism MUST have been performed as AT using CAN. 3. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism (Primary trust point, i.e. DO87). 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. '90 00' within a valid Secure Messaging response. 2. some OS dependent error within a valid Secure Messaging response.

3.8.7 Test case EAC2_ISO7816_M_7

Before performing this test case, validate, that the trust point has successfully been updated in test cases EAC2_ISO7816_M_3, EAC2_ISO7816_M_4 and EAC2_ISO7816_M_5.

Test - ID	EAC2_ISO7816_M_7
Purpose	Test the "Trust Point" update mechanism with two link certificates. This test changes the AT trust points.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. Validate that EAC2_ISO7816_M_3, EAC2_ISO7816_M_4 and EAC2_ISO7816_M_5 have been performed successfully 2. The PACE mechanism MUST have been performed as AT using CAN. 3. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference>

	• The Certification Authority Reference MUST be used as returned by the PACE mechanism (Primary trust point, i.e. DO87). 2. Send the appropriate link certificate as specified in the "Certificate Set 23" as LINK_CERT_23a. The eID Card MUST update the trust point with this new certificate. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <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 again and perform PACE again and verify that the new trust point is at the first position and the previous one has been moved to the second position 4. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the trust point received in DO 87 as returned by the PACE mechanism. 5. Send the appropriate link certificate as specified in the "Certificate Set 23" as LINK_CERT_23b. The eID Card MUST update the trust point with this new certificate. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> 6. 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 again and perform PACE again and verify that the new trust point is at the first position and the previous one has been moved to the second position 7. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the trust point received in DO 87 as returned by the PACE mechanism. 8. Send the appropriate DV-Certificate as specified in the "Certificate Set 23" as DV_CERT_23. '0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body>

	5F 37 <L_{5F37}> <certificate signature> • Since the trust point is still valid, the certificate MUST be verified successfully.
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. true 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. true 7. ’90 00’ within a valid Secure Messaging response. 8. ’90 00’ within a valid Secure Messaging response.

3.8.8 Test case EAC2_ISO7816_M_8

Before performing this test case, validate that the trust points have successfully been updated in test cases EAC2_ISO7816_M_3, EAC2_ISO7816_M_4, EAC2_ISO7816_M_5 and EAC2_ISO7816_M_7

Test - ID	EAC2_ISO7816_M_8
Purpose	Test the “Trust Point” update mechanism independence to other applications. The IS trust points MUST NOT be affected by AT trust point updates. This test works with IS trust points.
Version	EAC2_1.0
Profile	TA2
Preconditions	1. Validate that EAC2_ISO7816_M_3, EAC2_ISO7816_M_4, EAC2_ISO7816_M_5 and EAC2_ISO7816_M_7 have been performed successfully 2. The PACE mechanism MUST have been performed as IS using CAN. 3. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the trust point received in DO 87 as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 11” as DV_CERT_11d. ‘0C 2A 00 BE <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response.

3.9 Unit test EAC2_ISO7816_N – Migration policies

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 and trust point indicated by the passport provider. After the algorithm has been updated, the full test specification has to be repeated based on this new algorithm.

3.9.1 Test case EAC2_ISO7816_N_1

Test - ID	EAC2_ISO7816_N_1
Purpose	Test of the TA mechanism migration according to the manufacturer's implementation statement. Replacement of the IS trust point.
Version	EAC2_1.0
Profile	TA2, MIG
Preconditions	1. The PACE mechanism MUST have been performed. 2. All APDUs are sent as valid SecureMessaging APDUs. 3. This test case can only be done AFTER EAC2_ISO7816_M_5 has been performed.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be the Trust point received in DO 87 as returned by the PACE mechanism. 2. Send the appropriate link certificate with the updated mechanism as defined in "Certificate Set 13" as LINK_CERT_13. The ePassport MUST update the trust point with this new certificate. '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 eID Card. '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 reference> • The Certification Authority Reference MUST be used as specified in the Link certificate used in step 2. • The chip MUST 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 in the "Certificate Set 13" as.DV_CERT_13.

	‘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 DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as specified in the DV Certificate used in step 4. 6. Send the appropriate IS-Certificate as specified in the “Certificate Set 13” as IS_CERT_13 ‘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’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ’90 00’ within a valid Secure Messaging response.

3.9.2 Test case EAC2_ISO7816_N_2

Test - ID	EAC2_ISO7816_N_2
Purpose	Test of the TA mechanism migration according to the manufacturer’s implementation statement. Replacement of the AT trust point
Version	EAC2_1.0
Profile	TA2, MIG
Preconditions	Analog to EAC2_ISO7816_N_1, but with IS certificate chain
Test scenario	Analog to EAC2_ISO7816_N_1, but with IS certificate chain
Expected results	Analog to EAC2_ISO7816_N_1

3.10 Unit EAC2_ISO7816_O Effective Access Conditions with PACE CHAT Restrictions

This Unit extends Unit EAC2_ISO7816_L Effective Access Conditions. Most of the tests are repeated here, but the access is restricted by the CHAT submitted within the PACE mechanism.

3.10.1 Test case EAC2_ISO7816_O_1

Test - ID	EAC2_ISO7816_O_1
Purpose	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 but CHAT forbids access to DG3.
Version	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE mechanism MUST have been performed using CAN. The following CHAT MUST be used: '02' 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 3" chapter as DV_CERT_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 3 and 4. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 3" chapter as IS_CERT_3a. '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 eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference>

	83 <L_{83}> <Certificate Holder Reference > • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_03. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.10.2 Test case EAC2_ISO7816_O_2

Test – ID	EAC2_ISO7816_O_2
Purpose	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 but CHAT forbids access to DG4.
Version	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE mechanism MUST have been performed using CAN. The following CHAT MUST be used: '01' 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

- <Cryptogram> contains the following encrypted data objects
  `83 <L83> <Certification Authority Reference>`
- The Certification Authority Reference MUST be used as returned by the PACE mechanism.

2. Send the appropriate DV-Certificate as specified in the "Certificate Set 3" chapter as DV_CERT_3
   `'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>'`
   - <Cryptogram> contains the following encrypted data objects
     `7F 4E <L7F4E> <certificate body>`
     `5F 37 <L5F37> <certificate signature>`
   - This DV-Certificate grants access to data group 3 and 4.
3. Send the given MSE: Set DST APDU to the eID Card.
   `'0C 22 81 B6 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
   - <Cryptogram> contains the following encrypted data objects
     `83 <L83> <Certification Authority Reference>`
   - The Certificate 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 3" chapter as IS_CERT _3b.
   `'0C 2A 00 BE <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> <Le>'`
   - <Cryptogram> contains the following encrypted data objects
     `7F 4E <L7F4E> <certificate body>`
     `5F 37 <L5F37> <certificate signature>`
   - This IS-Certificate grants only access to data group 4.
5. Send the given MSE: Set AT APDU to the eID Card.
   `'0C 22 81 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`
   - <Cryptogram> contains the following encrypted data objects
     `80 <L80> <Cryptographic Mechanism Reference>`
     `83 <L83> <Certificate Holder Reference>`
   - The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used.
6. Send the given Get Challenge APDU to the eID Card.
   `'0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00'`
7. Send the given external authenticate command to the eID Card.
   `'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_03.
8. The Chip Authentication mechanism MUST be performed.
9. Send the given Select Application APDU to the eID Card (selecting ePassport application):
   `'0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'`

	• <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true 9. '90 00' within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.10.3 Test case EAC2_ISO7816_O_3

Test - ID	EAC2_ISO7816_O_3
Purpose	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 but CHAT forbids access to DG3.
Version	EAC2_1.0
Profile	ePassport, TA2, DG3
Preconditions	1. The PACE mechanism MUST have been performed using CAN. The following CHAT MUST be used: '02' 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 4" chapter as DV_CERT_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> • This DV-Certificate grants access to data group 3 only. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	• <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 4" chapter as IS_CERT_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> • This IS-Certificate grants access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '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_04. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. true

	9. ’90 00’ within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.10.4 Test case EAC2_ISO7816_O_4

Test - ID	EAC2_ISO7816_O_4
Purpose	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 enables access to both data 3 and 4, but CHAT forbids access to data group 4.
Version	EAC2_1.0
Profile	ePassport, TA2, DG4
Preconditions	1. The PACE mechanism MUST have been performed using CAN. The following CHAT MUST be used: ‘01’ 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the “Certificate Set 5” chapter as DV_CERT_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> • This DV-Certificate grants access to data group 4 only. 3. Send the given MSE: Set DST APDU to the eID Card. ‘0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate 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 5” chapter as IS_CERT_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> • This IS-Certificate grants access to data group 3 and 4. 5. Send the given MSE: Set AT APDU to the eID Card. ‘0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> • The Certificate Holder Reference stored inside the IS-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘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_05. 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting ePassport application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted ePassport application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, 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’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. true 9. ’90 00’ within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

3.10.5 Test case EAC2_ISO7816_O_5 Template

Test - ID	EAC2_ISO7816_O_5_template
Purpose	Test with a valid terminal authentication process with read access permission for well defined DGs if the DV certificate permits read access to all DGs while the Terminal certificate restricts access to one DG. DV certificate is an official domestic certificate. CHAT forbids access to the specific DG.
Version	See Table 14
Profile	eID, TA2, required data group presence see Table 14
Preconditions	1. The PACE mechanism MUST have been performed using PIN. See Table 14 for CHAT that has to be used

	2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 19" chapter as DV_CERT_19. '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 read access to all data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 19" chapter as defined in Table 14, column *Cert Reference* '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 Terminal-Certificate grants access to data groups as defined in Table 14, column *Access Rules*. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted terminal generated signature

	8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): '0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, to verify the read access to the selected data group has NOT been granted. '0C B0 (80 | <SFI>) 00 0D 97 01 01 8E 08 <Checksum> 00' • <SFI> contains the SFI reference as defined in Table 14, column *SFI*.
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '90 00' within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

Test case EAC2_ISO7816_O_5a to Test case EAC2_ISO7816_O_5v

Test Case ID	Version	Access Rules	Cert Reference	SFI
EAC2_ISO7816_O_5a	EAC2_1.0	This terminal certificate grants only read access to data group 1	AT_CERT_19a	0x01
EAC2_ISO7816_O_5b	EAC2_1.0	This terminal certificate grants only read access to data group 2	AT_CERT_19b	0x02
EAC2_ISO7816_O_5c	EAC2_1.0	This terminal certificate grants only read access to data group 3	AT_CERT_19c	0x03
EAC2_ISO7816_O_5d	EAC2_1.0	This terminal certificate grants only read access to data group 4	AT_CERT_19d	0x04
EAC2_ISO7816_O_5e	EAC2_1.0	This terminal certificate grants only read access to data group 5	AT_CERT_19e	0x05
EAC2_ISO7816_O_5f	EAC2_1.0	This terminal certificate grants only read access to data group 6	AT_CERT_19f	0x06
EAC2_ISO7816_O_5g	EAC2_1.0	This terminal certificate grants only read access to data group 7	AT_CERT_19g	0x07
EAC2_ISO7816_O_5h	EAC2_1.0	This terminal certificate grants only read access to data group 8	AT_CERT_19h	0x08
EAC2_ISO7816_O_5i	EAC2_1.0	This terminal certificate grants only read access to data group 9	AT_CERT_19i	0x09
EAC2_ISO7816_O_5j	EAC2_1.0	This terminal certificate grants only read access to data group 10	AT_CERT_19j	0x0a
EAC2_ISO7816_O_5k	EAC2_1.0	This terminal certificate grants only read access to data group 11	AT_CERT_19k	0x0b
EAC2_ISO7816_O_5l	EAC2_1.0	This terminal certificate grants only read access to data group 12	AT_CERT_19l	0x0c
EAC2_ISO7816_O_5m	EAC2_1.0	This terminal certificate grants only read access to data group 13	AT_CERT_19m	0x0d
EAC2_ISO7816_O_5n	EAC2_1.0	This terminal certificate grants only read access to data group 14	AT_CERT_19n	0x0e
EAC2_ISO7816_O_5o	EAC2_1.0	This terminal certificate grants only read access to data group 15	AT_CERT_19o	0x0f
EAC2_ISO7816_O_5p	EAC2_1.0	This terminal certificate grants only read access to data group 16	AT_CERT_19p	0x10
EAC2_ISO7816_O_5q	EAC2_1.0	This terminal certificate grants only read access to data group 17	AT_CERT_19q	0x11
EAC2_ISO7816_O_5r	EAC2_1.0	This terminal certificate grants only read access to data group 18	AT_CERT_19r	0x12
EAC2_ISO7816_O_5s	EAC2_1.0	This terminal certificate grants only read access to data group 19	AT_CERT_19s	0x13
EAC2_ISO7816_O_5t	EAC2_1.0	This terminal certificate grants only read access to data group 20	AT_CERT_19t	0x14
EAC2_ISO7816_O_5u	EAC2_1.0	This terminal certificate grants only read access to data group 21	AT_CERT_19u	0x15
EAC2_ISO7816_O_5v	EAC2_1.1	This terminal certificate grants only read access to data group 22	AT_CERT_19w	0x16

Table 14: Test cases EAC2_ISO7816_O_5

3.10.6 Test case EAC2_ISO7816_O_6 Template

Test - ID	EAC2_ISO7816_O_6_template
Purpose	Test with a valid terminal authentication process with read access permission for well defined DGs if the DV certificate permits read access to all DGs while the Terminal certificate restricts access to one DG. DV certificate is a non-official certificate. CHAT forbids access to the specific DG.
Version	See Table 15
Profile	eID, TA2, required data group presence see Table 15
Preconditions	1. The PACE mechanism MUST have been performed using PIN. See Table 15 for CHAT to be used 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 20" chapter as DV_CERT_20. '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 read access to all data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 20" chapter as defined in Table 15, column *Cert Reference* '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 Terminal-Certificate grants access to data groups as defined in Table 15, column *Access Rules* 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Read Binary (with SFI) command to the eID Card, to verify the read access to the selected data group has NOT been granted. ‘0C B0 (80 | <SFI>) 00 0D 97 01 01 8E 08 <Checksum> 00’ • <SFI> contains the SFI reference as defined in Table 15, column *SFI*.
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

Test case EAC2_ISO7816_O_6a to Test case EAC2_ISO7816_O_6v:

Test Case ID	Version	Access Rules	Cert Reference	SFI	CHAT
EAC2_ISO7816_O_6a	EAC2_1.0	This terminal certificate grants only read access to data group 1	AT_CERT_20a	0x01	'3E 1F FF FE F7'
EAC2_ISO7816_O_6b	EAC2_1.0	This terminal certificate grants only read access to data group 2	AT_CERT_20b	0x02	'3E 1F FF FD F7'
EAC2_ISO7816_O_6c	EAC2_1.0	This terminal certificate grants only read access to data group 3	AT_CERT_20c	0x03	'3E 1F FF FB F7'
EAC2_ISO7816_O_6d	EAC2_1.0	This terminal certificate grants only read access to data group 4	AT_CERT_20d	0x04	'3E 1F FF F7 F7'
EAC2_ISO7816_O_6e	EAC2_1.0	This terminal certificate grants only read access to data group 5	AT_CERT_20e	0x05	'3E 1F FF EF F7'
EAC2_ISO7816_O_6f	EAC2_1.0	This terminal certificate grants only read access to data group 6	AT_CERT_20f	0x06	'3E 1F FF DF F7'
EAC2_ISO7816_O_6g	EAC2_1.0	This terminal certificate grants only read access to data group 7	AT_CERT_20g	0x07	'3E 1F FF BF F7'
EAC2_ISO7816_O_6h	EAC2_1.0	This terminal certificate grants only read access to data group 8	AT_CERT_20h	0x08	'3E 1F FF 7F F7'
EAC2_ISO7816_O_6i	EAC2_1.0	This terminal certificate grants only read access to data group 9	AT_CERT_20i	0x09	'3E 1F FE FF F7'
EAC2_ISO7816_O_6j	EAC2_1.0	This terminal certificate grants only read access to data group 10	AT_CERT_20j	0x0a	'3E 1F FD FF F7'
EAC2_ISO7816_O_6k	EAC2_1.0	This terminal certificate grants only read access to data group 11	AT_CERT_20k	0x0b	'3E 1F FB FF F7'
EAC2_ISO7816_O_6l	EAC2_1.0	This terminal certificate grants only read access to data group 12	AT_CERT_20l	0x0c	'3E 1F F7 FF F7'
EAC2_ISO7816_O_6m	EAC2_1.0	This terminal certificate grants only read access to data group 13	AT_CERT_20m	0x0d	'3E 1F EF FF F7'
EAC2_ISO7816_O_6n	EAC2_1.0	This terminal certificate grants only read access to data group 14	AT_CERT_20n	0x0e	'3E 1F DF FF F7'
EAC2_ISO7816_O_6o	EAC2_1.0	This terminal certificate grants only read access to data group 15	AT_CERT_20o	0x0f	'3E 1F BF FF F7'
EAC2_ISO7816_O_6p	EAC2_1.0	This terminal certificate grants only read access to data group 16	AT_CERT_20p	0x10	'3E 1F 7F FF F7'
EAC2_ISO7816_O_6q	EAC2_1.0	This terminal certificate grants only read access to data group 17	AT_CERT_20q	0x11	'3E 1E FF FF F7'
EAC2_ISO7816_O_6r	EAC2_1.0	This terminal certificate grants only read access to data group 18	AT_CERT_20r	0x12	'3E 1D FF FF F7'
EAC2_ISO7816_O_6s	EAC2_1.0	This terminal certificate grants only read access to data group 19	AT_CERT_20s	0x13	'3E 1B FF FF F7'
EAC2_ISO7816_O_6t	EAC2_1.0	This terminal certificate grants only read access to data group 20	AT_CERT_20t	0x14	'3E 17 FF FF F7'
EAC2_ISO7816_O_6u	EAC2_1.0	This terminal certificate grants only read access to data group 21	AT_CERT_20u	0x15	'3E 0F FF FF F7'
EAC2_ISO7816_O_6v	EAC2_1.1	This terminal certificate grants only read access to data group 22	AT_CERT_20v	0x16	'3E 1F FF FF F7'

Table 15: Test cases EAC2_ISO7816_O_6

3.10.7 Test case EAC2_ISO7816_O_7 Template

Test - ID	EAC2_ISO7816_O_7_template
Purpose	Test with a valid terminal authentication process with write access permission for well defined DGs if the DV certificate permits write access to all writable DGs while the Terminal certificate restricts access to on DG. DV certificate is an official domestic certificate. CHAT forbids access to the specific DG
Version	See Table 16
Profile	eID, TA2, required data group presence see Table 16
Preconditions	1. The PACE mechanism MUST have been performed using PIN. See Table 16 for CHAT to be used. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 21" chapter as DV_CERT_21. '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 write access to all writable data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 21" chapter as referenced in Table 16, column *Cert Reference* '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 Terminal-Certificate grants access to data groups as defined in Table 16, column *Access Rules*. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Update Binary (with SFI) command to the eID Card, to verify that write access to the selected data group has NOT been granted. ‘0C D6 (80 | <SFI>) 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: 01 02 03 04 • <SFI> contains the SFI reference as defined in Table 16, column SFI.
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

Test case EAC2_ISO7816_O_7a to Test case EAC2_ISO7816_O_7f:

Test Case ID	Version	Access Rules	Cert Reference	SFI	CHAT
EAC2_ISO7816_O_7a	EAC2_1.0	This terminal certificate grants only write access to data group 17	AT_CERT_21a	0x11	‘1F 1F FF FF F7’
EAC2_ISO7816_O_7b	EAC2_1.0	This terminal certificate grants only write access to data group 18	AT_CERT_21b	0x12	‘2F 1F FF FF F7’
EAC2_ISO7816_O_7c	EAC2_1.0	This terminal certificate grants only write access to data group 19	AT_CERT_21c	0x13	’37 1F FF FF F7’
EAC2_ISO7816_O_7d	EAC2_1.0	This terminal certificate grants only write access to data group 20	AT_CERT_21d	0x14	‘3B 1F FF FF F7’
EAC2_ISO7816_O_7e	EAC2_1.0	This terminal certificate grants only write access to data group 21	AT_CERT_21e	0x15	‘3D 1F FF FF F7’
EAC2_ISO7816_O_7f	EAC2_1.1	This terminal certificate grants only write access to data group 22	AT_CERT_21f	0x16	‘3D 2F FF FF F7’

Table 16: Test cases EAC2_ISO7816_O_7

3.10.8 Test case EAC2_ISO7816_O_8 Template

Test - ID	EAC2_ISO7816_O_8_template
Purpose	Test with a valid terminal authentication process with write access permission for well defined DGs if the DV certificate permits write access to all writable DGs while the Terminal certificate restricts access to one DG. DV certificate is a non-official certificate. CHAT forbids access to the specific DG.
Version	See Table 17
Profile	eID, TA2, required data group presence see Table 17
Preconditions	1. The PACE mechanism MUST have been performed using PIN. See Table 17 for CHAT to be used. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 22" chapter as DV_CERT_22. '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 write access to all writable data groups. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 22" chapter as defined in Table 17, column *Cert Reference*. '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 Terminal-Certificate grants only write access to data groups as defined in Table 17, column *Access Rules*. 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Update Binary (with SFI) command to the eID Card, to verify that write access to the selected data group has NOT been granted. ‘0C D6 (80 | <SFI>) 00 0D <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: 01 02 03 04 • <SFI> contains the SFI reference as defined in Table 17, column *SFI*.
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. Checking error within a valid Secure Messaging response.

Test case EAC2_ISO7816_O_8a to Test case EAC2_ISO7816_O_8f:

Test Case ID	Version	Access Rules	Cert Reference	SFI	CHAT
EAC2_ISO7816_O_8a	EAC2_1.0	This terminal certificate grants only write access to data group 17	AT_CERT_22a	0x11	‘1F 1F FF FF F7’
EAC2_ISO7816_O_8b	EAC2_1.0	This terminal certificate grants only write access to data group 18	AT_CERT_22b	0x12	‘2F 1F FF FF F7’
EAC2_ISO7816_O_8c	EAC2_1.0	This terminal certificate grants only write access to data group 19	AT_CERT_22c	0x13	’37 1F FF FF F7’
EAC2_ISO7816_O_8d	EAC2_1.0	This terminal certificate grants only write access to data group 20	AT_CERT_22d	0x14	‘3B 1F FF FF F7’
EAC2_ISO7816_O_8e	EAC2_1.0	This terminal certificate grants only write access to data group 21	AT_CERT_22e	0x15	‘3D 1F FF FF F7’
EAC2_ISO7816_O_8f	EAC2_1.1	This terminal certificate grants only write access to data group 22	AT_CERT_22f	0x16	‘3D 2F FF FF F7’

Table 17: Test cases EAC2_ISO7816_O_8

3.10.9 Test case EAC2_ISO7816_O_9

Test - ID	EAC2_ISO7816_O_9
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is Age Verification. CHAT forbids age verification.
Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed using PIN. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17f. '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 Terminal-Certificate grants access to special function "Age Verification" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00’ • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 67 <L_{67}> <Auxiliary Data> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. • Auxiliary Data contains valid Date of Birth data. 6. Send the given Get Challenge APDU to the eID Card. ‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ’69 82’ within a valid Secure Messaging response.

3.10.10 Test case EAC2_ISO7816_O_10

Test - ID	EAC2_ISO7816_O_10
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is Municipality ID Verification. CHAT forbids Municipality ID Verification.

Version	EAC2_1.0
Profile	eID, TA2
Preconditions	1. The PACE mechanism MUST have been performed using PIN. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17g. '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 Terminal-Certificate grants access to special function "Age Verification" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference > 67 <L_{67}> <Auxiliary Data> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. • Auxiliary Data contains valid Municipality ID data. 6. Send the given Get Challenge APDU to the eID Card.

	‘0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00’ 7. Send the given external authenticate command to the eID Card. ‘0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Select Application APDU to the eID Card (selecting eID application): ‘0C A4 04 0C <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted eID application-ID. 10. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response. 4. ’90 00’ within a valid Secure Messaging response. 5. ’90 00’ within a valid Secure Messaging response. 6. ‘<Eight bytes of random data> 90 00’ within a valid Secure Messaging response. 7. ’90 00’ within a valid Secure Messaging response. 8. True 9. ’90 00’ within a valid Secure Messaging response. 10. ’69 82’. The error MUST be encoded in a valid Secure Messaging response.

3.10.11 Test case EAC2_ISO7816_O_11

Test - ID	EAC2_ISO7816_O_11
Version	deleted in version 1.00 RC

3.10.12 Test case EAC2_ISO7816_O_12

Test - ID	EAC2_ISO7816_O_12
Purpose	Test with a valid terminal authentication process with rights for special functions if the DV certificate permits all special functions while the terminal certificate restricts access to one special function. DV certificate is an official domestic certificate. Special function allowed by terminal certificate is “PIN Management”. CHAT forbids “PIN Management”
Version	EAC2_1.0
Profile	eID, TA2

Preconditions	1. The PACE mechanism MUST have been performed using PIN. 2. All APDUs are sent as valid SecureMessaging APDUs.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the appropriate DV-Certificate as specified in the "Certificate Set 17" chapter as DV_CERT_17. '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 all eID special functions. 3. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 4. Send the appropriate Terminal-Certificate as specified in the "Certificate Set 17" chapter as AT_CERT_17b. '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 Terminal-Certificate grants access to special function "PIN Management" 5. Send the given MSE: Set AT APDU to the eID Card. '0C 22 81 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 80 <L_{80}> <Cryptographic Mechanism Reference> 83 <L_{83}> <Certificate Holder Reference> • The Certificate Holder Reference stored inside the Terminal-Certificate sent in step 4 has to be used. 6. Send the given Get Challenge APDU to the eID Card. '0C 84 00 00 0D 97 01 08 8E 08 <Checksum> 00' 7. Send the given external authenticate command to the eID Card. '0C 82 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'

	• <Cryptogram> contains the encrypted terminal generated signature 8. The Chip Authentication mechanism MUST be performed. 9. Send the given Deactivate PIN APDU to the eID Card: '0C 04 10 03 <Lc> 8E 08 <Checksum> 00'
Expected results	1. '90 00' within a valid Secure Messaging response. 2. '90 00' within a valid Secure Messaging response. 3. '90 00' within a valid Secure Messaging response. 4. '90 00' within a valid Secure Messaging response. 5. '90 00' within a valid Secure Messaging response. 6. '<Eight bytes of random data> 90 00' within a valid Secure Messaging response. 7. '90 00' within a valid Secure Messaging response. 8. True 9. '69 82'. The error MUST be encoded in a valid Secure Messaging response.

3.11 Unit test EAC2_ISO7816_P – PIN-Management

This unit covers all tests about PINs. PINs are used for the ePassport, eID and eSign application. [R8] defines 4 types of PINs used in different contexts.

- **CAN:** The Card Access Number (CAN) is a short password that is printed or displayed on the document.
- **PIN:** The Personal Identification Number (PIN) is a short secret password that SHALL be only known to the legitimate holder of the document.
- **PUK:** The PIN Unblock Key (PUK) is a long secret password that SHALL be only known to the legitimate holder of the document.
- **MRZ:** The MRZ-Password is a secret key that is derived from the machine readable zone and may be used for both PACE and BAC.

3.11.1 Test case EAC2_ISO7816_P_1

Test – ID	EAC2_ISO7816_P_1
Purpose	Reduce initial PIN retry counter by 1
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. PIN retry counter MUST be set to initial value 3. Use INVALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm.

	• The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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. Power off the chip and reinitialize connection 7. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
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. '63 00' or '63 CX' where X indicates the number of remaining verification tries, i.e. initial value – 1 (see ICS). 6. TRUE 7. '63 CX' where X indicates the number of remaining verification tries, i.e. initial value – 1 (see ICS).

3.11.2 Test case EAC2_ISO7816_P_2

Test – ID	EAC2_ISO7816_P_2
Purpose	Reset PIN retry counter to initial value
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended

	2. This test case MUST be performed immediately after Test case EAC2_ISO7816_P_1. 3. Use VALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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. Power off the chip and reinitialize connection 7. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
Expected results	1. '63 CX' where X indicated the number of remaining verification tries, i.e. initial value – 1 (see ICS). 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. TRUE 7. '90 00'

3.11.3 Test case EAC2_ISO7816_P_3

Test – ID	EAC2_ISO7816_P_3
Purpose	Suspend PIN
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. Use INVALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card to map the nonce: '10 86 00 00 <Lc> 81 <L_{81}> <mapping data> <Le>' 4. Perform key agreement: '10 86 00 00 <Lc> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 85 <L_{85}> <authentication token> <Le>' 6. Power off the chip and reinitialize connection 7. Go to step 1 and repeat all steps until step 1 returns '63 C1'
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. '63 00' or '63 CX' where X indicates the number of remaining verification tries. 6. TRUE 7. '63 CX'. Repeat until X=1. The PICC MUST reduce X by 1 on each run.

3.11.4 Test case EAC2_ISO7816_P_4

Test – ID	EAC2_ISO7816_P_4
Purpose	PIN Authentication attempt with suspended PIN

Version	EAC2_1.0
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked or deactivated 2. The PIN MUST have been suspended (e.g. using Test case EAC2_ISO7816_P_3)
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
Expected results	1. '63 C1'

3.11.5 Test case EAC2_ISO7816_P_5

Test – ID	EAC2_ISO7816_P_5
Purpose	CAN Authentication attempt with suspended PIN, resume with PIN
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked or deactivated 2. The PIN MUST have been suspended (e.g. using Test case EAC2_ISO7816_P_3)
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with CAN: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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. Send the given MSE: Set AT APDU to the eID Card. '0C 22 C1 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: 7C <L_{7C}> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain> • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 7. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '1C 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 00' 8. Send the given General Authenticate APDU to the eID Card. '1C 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}> <mapping data>' 9. Send the given General Authenticate APDU to the eID Card. '1C 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}> 83 <L_{83}> <ephemeral public key>' 10. Send the given General Authenticate APDU to the eID Card. '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}> 85 <L_{81}> <authentication token>' 11. Power off the chip and reinitialize connection 12. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored

	in EF.CardAccess.
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. ’63 C1’ within a valid SM response 7. 7C <L_{7C}> ‘80’ <L_{80}> <encrypted nonce> ’90 00’ within a valid SM response 8. 7C <L_{7C}> ‘82’ <L_{82}> <mapping data> ’90 00’ within a valid SM response 9. 7C <L_{7C}> ‘84’ <L_{84}> <ephemeral public key> ’90 00’ within a valid SM response 10. 7C <L_{7C}> ‘86’ <L_{86}> <authentication token> ’90 00’ within a valid SM response 11. TRUE 12. ’90 00’

3.11.6 Test case EAC2_ISO7816_P_6

Test – ID	EAC2_ISO7816_P_6
Purpose	Check volatile resumed status of PIN using PACE with CAN
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked or deactivated 2. The PIN MUST have been suspended (e.g. using Test case EAC2_ISO7816_P_3)
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with CAN: ’00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>’ • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: ’10 86 00 00 <Lc> 7C 00 <Le>’ 3. Send the given General Authenticate APDU to the eID Card 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. Power off the chip and reinitialize connection 7. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
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. TRUE 7. '63 C1'

3.11.7 Test case EAC2_ISO7816_P_7

Test – ID	EAC2_ISO7816_P_7
Purpose	Change PIN
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. Use VALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the

	encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '<new PIN>' 7. Power off the chip and reinitialize connection 8. Perform PACE to verify new PIN, e.g. using Test case EAC2_ISO7816_H_2
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. '90 00' within a valid SM response 7. TRUE 8. TRUE

3.11.8 Test case EAC2_ISO7816_P_8

Test – ID	EAC2_ISO7816_P_8
Purpose	Block PIN
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked or deactivated 2. The PIN MUST have been suspended (e.g. using Test case EAC2_ISO7816_P_3) 3. Use INVALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using CAN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 02 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm.

- The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.

2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce:
   '10 86 00 00 <Lc> 7C 00 <Le>'
3. Send the given General Authenticate APDU to the eID Card 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. Send the given MSE: Set AT APDU to the eID Card.
   '0C 22 C1 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00'
   - <Cryptogram> contains the following encrypted data objects:
     7C <L_{7C}>
     80 <L_{80}> <PACE OID>
     83 01 03
     84 <L_{84}> <PACE domain>
   - PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm.
   - The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
7. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce:
   '1C 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 00'
8. Send the given General Authenticate APDU to the eID Card.
   '1C 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}> <mapping data>'
9. Send the given General Authenticate APDU to the eID Card.
   '1C 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}> 83 <L_{81}> <ephemeral public key>'
10. Send the given General Authenticate APDU to the eID Card.

	'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}> 85 <L_{81}> <authentication token>'
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. '63 C1' within a valid SM response 7. 7C <L_{7C}> '80' <L_{80}> <encrypted nonce> '90 00' within a valid SM response 8. 7C <L_{7C}> '82' <L_{82}> <mapping data> '90 00' within a valid SM response 9. 7C <L_{7C}> '84' <L_{84}> <ephemeral public key> '90 00' within a valid SM response 10. '63 00' or '63 C0' within a valid SM response

3.11.9 Test case EAC2_ISO7816_P_8a

Test – ID	EAC2_ISO7816_P_8a
Purpose	PIN Authentication attempt with blocked PIN
Version	EAC2_1.0, moved Test case EAC2_ISO7816_H_21
Profile	PACE
Preconditions	1. The PIN MUST NOT have been deactivated 2. The PIN MUST have been blocked (e.g. using Test case EAC2_ISO7816_P_8)
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with PIN: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
Expected results	1. '63 C0'

3.11.10 Test case EAC2_ISO7816_P_9

Test – ID	EAC2_ISO7816_P_9
Purpose	Unblock PIN, use old PIN
Version	EAC2_1.02
Profile	PACE

Preconditions	1. The PIN MUST have been blocked (e.g. using Test case EAC2_ISO7816_P_8) 2. Use VALID PUK for key derivation process 3. Use OLD PIN after unblock mechanism
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PUK mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 04 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 03 03 0D 97 01 01 8E 08 <Checksum> 00' 7. Power off the chip and reinitialize connection 8. Send the given MSE: Set AT APDU to the eID Card using OLD PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
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. '90 00' within a valid SM response

	7. TRUE 8. ’90 00’

3.11.11 Test case EAC2_ISO7816_P_10

Test – ID	EAC2_ISO7816_P_10
Purpose	Unblock PIN, use NEW PIN
Version	EAC2_1.02
Profile	PACE, CNG_PIN_PUK
Preconditions	1. The PIN MUST have been blocked(e.g. using Test case EAC2_ISO7816_P_8) 2. Use VALID PUK for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PUK mechanism: ’00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 04 84 <L_{84}> <PACE domain>’ • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: ’10 86 00 00 <Lc> 7C 00 <Le>’ 3. Send the given General Authenticate APDU to the eID Card 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. Send the given Reset Retry Counter APDU to the eID Card. ‘0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: ‘<new PIN>’ 7. Power off the chip and reinitialize connection 8. Perform PACE to verify new PIN, e.g. using Test case EAC2_ISO7816_H_2
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. '90 00' within a valid SM response 7. TRUE 8. TRUE

3.11.12 Test case EAC2_ISO7816_P_11

Test – ID	EAC2_ISO7816_P_11
Purpose	Change PIN, PUK Authentication
Version	EAC2_1.02
Profile	PACE, CNG_PIN_PUK
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. Use VALID PUK for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PUK mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 04 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '<new PIN>' 7. Power off the chip and reinitialize connection 8. Perform PACE to verify new PIN, e.g. using Test case EAC2_ISO7816_H_2

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. ’90 00’ within a valid SM response 7. TRUE 8. TRUE

3.11.13 Test case EAC2_ISO7816_P_12

Test – ID	EAC2_ISO7816_P_12
Purpose	Negative test: Change PIN, PUK Authentication
Version	EAC2_1.02
Profile	PACE, NOT CNG_PIN_PUK
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. Use VALID PUK for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PUK mechanism: ’00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 04 84 <L_{84}> <PACE domain>’ • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: ’10 86 00 00 <Lc> 7C 00 <Le>’ 3. Send the given General Authenticate APDU to the eID Card 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. Send the given Reset Retry Counter APDU to the eID Card. ‘0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: ‘<new PIN>’

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. ’69 82’. The error MUST be encoded in a valid Secure Messaging response.

3.11.14 Test case EAC2_ISO7816_P_13

Test – ID	EAC2_ISO7816_P_13
Purpose	Negative test: Unblock PIN, use NEW PIN
Version	EAC2_1.02
Profile	PACE, NOT CNG_PIN_PUK
Preconditions	1. The PIN MUST have been blocked(e.g. using Test case EAC2_ISO7816_P_8) 2. Use VALID PUK for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PUK mechanism: ’00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 04 84 <L_{84}> <PACE domain>’ • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: ’10 86 00 00 <Lc> 7C 00 <Le>’ 3. Send the given General Authenticate APDU to the eID Card 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. Send the given Reset Retry Counter APDU to the eID Card. ‘0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: ‘<new PIN>’

	7. Power off the chip and reinitialize connection 8. Send the given MSE: Set AT APDU to the eID Card using OLD PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
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. '69 82'. The error MUST be encoded in a valid Secure Messaging response. 7. TRUE 8. '63 C0'. PIN MUST still be blocked.

3.11.15 Test case EAC2_ISO7816_P_14

Test – ID	EAC2_ISO7816_P_14
Purpose	Change PIN
Version	EAC2_1.02
Profile	PACE
Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. Use VALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 85 <L_{85}> <authentication token> <Le>' 6. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '<new PIN>' 7. Power off the chip and reinitialize connection 8. Perform PACE to verify new PIN, e.g. using Test case EAC2_ISO7816_H_2
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. '90 00' within a valid SM response 7. TRUE 8. TRUE

3.11.16 Test case EAC2_ISO7816_P_15

Test – ID	EAC2_ISO7816_P_15
Purpose	Change PIN via authenticated PIN management
Version	EAC2_1.0
Profile	PACE, TA2, CA2, CNG_PIN_AR
Preconditions	1. The PACE mechanism MUST have been performed using PIN, PIN management must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17b) 3. The Chip Authentication MUST have been performed 4. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects: '<new PIN>' 2. Power off the chip and reinitialize connection 3. Perform PACE to verify new PIN, e.g. using Test case EAC2_ISO7816_H_2

Expected results	1. ’90 00’ within a valid SM response 2. TRUE 3. TRUE

3.11.17 Test case EAC2_ISO7816_P_16

Test – ID	EAC2_ISO7816_P_16
Purpose	Change PIN via authenticated PIN management
Version	EAC2_1.0
Profile	PACE, TA2, CA2, NOT CNG_PIN_AR
Preconditions	1. The PACE mechanism MUST have been performed using PIN, PIN management must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17b) 3. The Chip Authentication MUST have been performed 4. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Reset Retry Counter APDU to the eID Card. ‘0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the following encrypted data objects: ‘<new PIN>’
Expected results	1. ’69 82’ within a valid SM response

3.11.18 Test case EAC2_ISO7816_P_17

Test – ID	EAC2_ISO7816_P_17
Purpose	Change CAN
Version	EAC2_1.0
Profile	PACE, TA2, CA2, CNG_CAN_AR
Preconditions	1. The PACE mechanism MUST have been performed using PIN, PIN management must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17b) 3. The Chip Authentication MUST have been performed 4. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Reset Retry Counter APDU to the eID Card. ‘0C 2C 02 02 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the following encrypted data objects: ‘<new CAN>’ 2. Power off the chip and reinitialize connection 3. Perform PACE to verify new CAN, e.g. using Test case EAC2_ISO7816_H_1
Expected results	1. ’90 00’ within a valid SM response

	2. TRUE 3. TRUE

3.11.19 Test case EAC2_ISO7816_P_18

Test – ID	EAC2_ISO7816_P_18
Purpose	Change CAN
Version	EAC2_1.0
Profile	PACE, TA2, CA2, NOT CNG_CAN_AR
Preconditions	1. The PACE mechanism MUST have been performed using PIN, PIN management must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17b) 3. The Chip Authentication MUST have been performed 4. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 02 02 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the following encrypted data objects: '<new CAN>'
Expected results	1. '69 82' within a valid SM response

3.11.20 Test case EAC2_ISO7816_P_19

Test – ID	EAC2_ISO7816_P_19
Purpose	Test with deactivated PIN
Version	EAC2_1.0, moved Test case EAC2_ISO7816_H_28
Profile	PACE
Preconditions	1. The PIN MUST have been deactivated
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card with PIN: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess.
Expected results	1. '62 83'

3.11.21 Test case EAC2_ISO7816_P_20

Test – ID	EAC2_ISO7816_P_20
Purpose	Try to change PIN, but NEW PIN is too short
Version	EAC2_1.02
Profile	PACE

Preconditions	1. The PIN MUST NOT have been blocked, deactivated or suspended 2. Use VALID PIN for key derivation process
Test scenario	1. Send the given MSE: Set AT APDU to the eID Card using PIN mechanism: '00 22 C1 A4 <Lc> 80 <L_{80}> <PACE OID> 83 01 03 84 <L_{84}> <PACE domain>' • PACE OID is a valid PACE OID (e.g. id-PACE-DH-3DES-CBC-CBC) fitting the implemented algorithm. • The PACE domain parameter reference is REQUIRED if the domain parameters are ambiguous, i.e. more than one set of domain parameters are available for PACE. The domain parameters are stored in EF.CardAccess. 2. Send the given General Authenticate APDU to the eID Card to get the encrypted nonce: '10 86 00 00 <Lc> 7C 00 <Le>' 3. Send the given General Authenticate APDU to the eID Card 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> 83 <L_{83}> <ephemeral public key> <Le>' 5. Perform mutual authentication: '00 86 00 00 <Lc> 85 <L_{85}> <authentication token> <Le>' 6. Send the given Reset Retry Counter APDU to the eID Card. '0C 2C 02 03 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '<new PIN>' • NEW PIN MUST be shorter than minimum PIN length stated in ICS 7. Power off the chip and reinitialize connection 8. Perform PACE to verify OLD PIN is still valid, e.g. using Test case EAC2_ISO7816_H_2
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. '69 82' or other error within a valid SM response 7. TRUE 8. TRUE

3.12 Unit test EAC2_ISO7816_Q Auxiliary Data Verification

This unit covers all tests about eID special functions “auxiliary data verification”, i. e. age verification, document validity verification and Municipality ID verification.

3.12.1 Test case EAC2_ISO7816_Q_1

Test – ID	EAC2_ISO7816_Q_1
Purpose	Positive age verification test, verification successful, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17f) 3. Auxiliary data with valid Date of Birth data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. The date of birth MUST fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. ’90 00’ within a valid Secure Messaging response.

3.12.2 Test case EAC2_ISO7816_Q_2

Test – ID	EAC2_ISO7816_Q_2
Purpose	Positive age verification test, verification fails, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17f) 3. Auxiliary data with valid Date of Birth data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. The date of birth MUST NOT fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs

Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '63 00'. The error MUST be encoded in a valid Secure Messaging response.

3.12.3 Test case EAC2_ISO7816_Q_3

Test – ID	EAC2_ISO7816_Q_3
Purpose	Age verification test with unauthorized terminal, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17a) 3. Auxiliary data with valid Date of Birth data object MUST have been sent by unauthorized terminal during Terminal Authentication mechanism. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '69 82'. The error MUST be encoded in a valid Secure Messaging response.

3.12.4 Test case EAC2_ISO7816_Q_4

Test – ID	EAC2_ISO7816_Q_4
Purpose	Age verification test with authorized terminal but without auxiliary data transmission, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed without optional transmission of auxiliary data (DV_CERT_17, AT_CERT_17f) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected

	5. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '6A 88'. The error MUST be encoded in a valid Secure Messaging response.

3.12.5 Test case EAC2_ISO7816_Q_5

Test – ID	EAC2_ISO7816_Q_5
Purpose	Positive age verification test, verification successful, non-official certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_18, AT_CERT_18f) 3. Auxiliary data with valid Date of Birth data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Date of birth MUST fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '90 00' within a valid Secure Messaging response

3.12.6 Test case EAC2_ISO7816_Q_6

Test – ID	EAC2_ISO7816_Q_6
Purpose	Positive document validity verification test, verification successful, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17g) 3. Auxiliary data with valid Document Validity data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Its

	date MUST fit document validity, i.e. <expiration date>-1 . 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry>
Expected results	1. '90 00' within a valid Secure Messaging response.

3.12.7 Test case EAC2_ISO7816_Q_7

Test – ID	EAC2_ISO7816_Q_7
Purpose	Document validity verification test, verification fails, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17g) 3. Auxiliary data with valid Document Validity data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Its date MUST NOT fit document validity, i.e. <expiration date>+1. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry>
Expected results	1. '63 00'. The error MUST be encoded in a valid Secure Messaging response.

3.12.8 Test case EAC2_ISO7816_Q_8

Test – ID	EAC2_ISO7816_Q_8
Purpose	Document Validity verification test with authorized terminal but without auxiliary data transmission, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN 2. The Terminal Authentication mechanism MUST have been performed without optional transmission of auxiliary data (DV_CERT_17,

	AT_CERT_17g) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry>
Expected results	1. '6A 88'. The error MUST be encoded in a valid Secure Messaging response.

3.12.9 Test case EAC2_ISO7816_Q_9

Test – ID	EAC2_ISO7816_Q_9
Purpose	Positive Document Validity verification test, verification successful, non-official certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_18, AT_CERT_18g) 3. Auxiliary data with valid Document Validity data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Document Validity MUST include the current date. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry>
Expected results	1. '90 00' within a valid Secure Messaging response.

3.12.10 Test case EAC2_ISO7816_Q_10

Test – ID	EAC2_ISO7816_Q_10
Purpose	Positive Municipality ID verification test, verification successful, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT

	2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17g) 3. Auxiliary data with valid Municipality ID data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Municipality ID MUST fit the required ID. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '90 00' within a valid Secure Messaging response.

3.12.11 Test case EAC2_ISO7816_Q_11

Test – ID	EAC2_ISO7816_Q_11
Purpose	MunicipalityID verification test, verification fails, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17g) 1. Auxiliary data with valid MunicipalityID data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. MunicipalityID MUST NOT fit the required ID. 2. The Chip Authentication MUST have been performed 3. The eID application MUST have been selected 4. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '63 00'. The error MUST be encoded in a valid Secure Messaging response.

3.12.12 Test case EAC2_ISO7816_Q_12

Test – ID	EAC2_ISO7816_Q_12
Purpose	MunicipalityID verification test with unauthorized terminal, official domestic certificate
Version	EAC2_1.0

Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17a) 3. Auxiliary data with valid MunicipalityID data object MUST have been sent by unauthorized terminal during Terminal Authentication mechanism. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '69 82'. The error MUST be encoded in a valid Secure Messaging response.

3.12.13 Test case EAC2_ISO7816_Q_13

Test – ID	EAC2_ISO7816_Q_13
Purpose	MunicipalityID verification test with authorized terminal but without auxiliary data transmission, official domestic certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed without optional transmission of auxiliary data (DV_CERT_17, AT_CERT_17g) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '6A 88'. The error MUST be encoded in a valid Secure Messaging response.

3.12.14 Test case EAC2_ISO7816_Q_14

Test – ID	EAC2_ISO7816_Q_14
Purpose	Positive MunicipalityID verification test, verification successful, non-official

	certificate
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_18, AT_CERT_18g) 3. Auxiliary data with valid MunicipalityID data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. MunicipalityID MUST fit the required ID. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '90 00' within a valid Secure Messaging response

3.12.15 Test case EAC2_ISO7816_Q_15

Test – ID	EAC2_ISO7816_Q_15
Purpose	Positive Municipality ID verification test, verification successful, official domestic certificate, check leftmost part of Municipality ID
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17g) 3. Auxiliary data with valid Municipality ID data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. MunicipalityID is truncated but the leftmost bytes MUST fit the required ID. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '90 00' within a valid Secure Messaging response.

3.12.16 Test case EAC2_ISO7816_Q_16

Test – ID	EAC2_ISO7816_Q_16
Purpose	Positive MunicipalityID verification test, verification successful, non-official certificate, check leftmost part of Municipality ID
Version	EAC2_1.0
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, Municipality ID verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_18, AT_CERT_18g) 3. Auxiliary data with valid MunicipalityID data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. MunicipalityID is truncated but the leftmost bytes MUST fit the required ID. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. '90 00' within a valid Secure Messaging response

3.12.17 Test case EAC2_ISO7816_Q_17

Test – ID	EAC2_ISO7816_Q_17
Purpose	Positive test with two Discretionary Data Templates in Authentication Data Object containing age verification data and document validity verification data, The verification must be successful with an official domestic certificate
Version	EAC2_1.1
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17f) 3. Auxiliary data with valid Date of Birth data object and document validity data object MUST have been sent by an authorized terminal during Terminal Authentication mechanism. The date of birth and document validity MUST fit the required age and validity. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card.

	‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth> 2. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response.

3.12.18 Test case EAC2_ISO7816_Q_18

Test – ID	EAC2_ISO7816_Q_18
Purpose	Positive test with three Discretionary Data Templates in Authentication Data Object containing age verification data, document validity verification data and MunicipalityID verification data. The verification must be successful with an official domestic certificate
Version	EAC2_1.1
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17j) 3. Auxiliary data with valid Date of Birth data object, document validity data object and MunicipalityID data object MUST have been sent by an authorized terminal during Terminal Authentication mechanism. The date of birth, document validity and MunicipalityID MUST fit the required age, validity and MunicipalityID. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth> 2. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry> 3. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’

	• <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response.

3.12.19 Test case EAC2_ISO7816_Q_19

Test – ID	EAC2_ISO7816_Q_19
Purpose	Positive test with two Discretionary Data Templates of same type in Authentication Data Object containing two age verification data objects with same OID. The verification of last the data object must be successful with an official domestic certificate
Version	EAC2_1.1
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed (DV_CERT_17, AT_CERT_17f) 3. Auxiliary data with two valid Date of Birth data objects MUST have been sent by an authorized terminal during Terminal Authentication mechanism. The first date of birth MUST NOT fit the required age, the second date of birth MUST fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. ‘8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. ’90 00’ within a valid Secure Messaging response.

3.12.20 Test case EAC2_ISO7816_Q_20

Test – ID	EAC2_ISO7816_Q_20
Purpose	Negative test with two Discretionary Data Templates of the same type in the Authentication Data Object containing two age verification data objects with the same OID. The verification of last the data object must fail with an official domestic certificate
Version	EAC2_1.1
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed

	(DV_CERT_17, AT_CERT_17f) 3. Auxiliary data with two valid Date of Birth data objects MUST have been sent by an authorized terminal during Terminal Authentication mechanism. The first date of birth MUST fit the required age, the second date of birth MUST NOT fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. Checking error in a valid Secure Messaging response.

3.12.21 Test case EAC2_ISO7816_Q_21

Test – ID	EAC2_ISO7816_Q_21
Purpose	Age verification test with authorized terminal but invalid auxiliary data object (wrong tag) and an official domestic certificate
Version	EAC2_1.1
Profile	eID, AUX
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with invalid (wrong tag 0x72 instead of 0x73) auxiliary data (DV_CERT_17, AT_CERT_17f) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '6A 88'. The error MUST be encoded in a valid Secure Messaging response.

3.12.22 Test case EAC2_ISO7816_Q_22

Test – ID	EAC2_ISO7816_Q_22
Purpose	Age verification test with authorized terminal but invalid auxiliary data object wrong OID tag) and an official domestic certificate
Version	EAC2_1.1
Profile	eID, AUX

Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with invalid (wrong tag 0x07 instead of 0x06 for OID) auxiliary data (DV_CERT_17, AT_CERT_17f) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Verify APDU to the eID Card. '8C 20 80 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '6A 88'. The error MUST be encoded in a valid Secure Messaging response.

3.13 Unit test EAC2_ISO7816_R Restricted Identification

This unit covers all tests about eID special function "restricted identification".
Note: This test unit has to be performed for each key specified in ICS.

3.13.1 Test case EAC2_ISO7816_R_1

Test – ID	EAC2_ISO7816_R_1
Purpose	Positive test for Restricted Identification, official domestic certificate
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_17, AT_CERT_17c) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '80' <L_{80}> <id-RI-x> '84' <L_{84}> <RefKeyID> 2. Send the given General Authenticate APDU to the eID Card: '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the sector public key PK_{Sector}

	7C <L_{7C}> 'A0' <L_{A0}> <PK_{Sector}>, Hash(PK_{Sector}) MUST fit the hash value encoded in AT_CERT_17c
Expected results	1. '90 00' within a valid Secure Messaging response 2. 7C <L_{7C}> '81' <L_{81}> <$I_{SectorPICC}$> '90 00' in valid Secure Messaging response

3.13.2 Test case EAC2_ISO7816_R_2

Test – ID	EAC2_ISO7816_R_2
Version	deleted in version 1.00 RC

3.13.3 Test case EAC2_ISO7816_R_3

Test – ID	EAC2_ISO7816_R_3
Purpose	Test for Restricted Identification with unauthorized terminal , official domestic certificate
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must NOT be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_17, AT_CERT_17c) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '80' <L_{80}> <id-RI-x> '84' <L_{84}> <RefKeyID> 2. Send the given General Authenticate APDU to the eID Card: '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the sector public key PK_{Sector} 7C <L_{7C}> 'A0' <L_{A0}> <PK_{Sector}>, Hash(PK_{Sector}) MUST fit the hash value encoded in AT_CERT_17c
Expected results	1. '90 00' within a valid Secure Messaging response. 2. expected result is CONDITIONAL: **For private keys with "authorized only" attribute set:** '69 82'. The error MUST be encoded in a valid Secure Messaging response. **For private keys with "authorized only" attribute NOT set:** 7C <L_{7C}> '81' <L_{81}> <$I_{SectorPICC}$> '90 00' in valid Secure Messaging response

3.13.4 Test case EAC2_ISO7816_R_4

Test – ID	EAC2_ISO7816_R_4
Version	deleted in version 1.00 RC

3.13.5 Test case EAC2_ISO7816_R_5

Test – ID	EAC2_ISO7816_R_5
Purpose	Test for Restricted Identification with unsupported algorithm, official domestic Certificate
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_17, AT_CERT_17c) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: `'0C 22 41 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08 <Checksum> 00'` • <Cryptogram> contains the following encrypted data objects: `'80' <L80> <BadOID> '84' <L84> <RefKeyID>` (Use 0.4.0.127.0.7.2.2.2.5.1 as BadOID)
Expected results	1. Checking error in valid Secure Messaging response

3.13.6 Test case EAC2_ISO7816_R_6

Test – ID	EAC2_ISO7816_R_6
Purpose	Test for Restricted Identification with invalid sector public key, official domestic Certificate
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_17, AT_CERT_17c) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: `'0C 22 41 A4 <Lc> 87 <L87> 01 <Cryptogram> 8E 08`

	<Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '80' <L_{80}> <id-RI-x> '84' <L_{84}> <RefKeyID> 2. Send the given General Authenticate APDU to the eID Card: '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' • <Cryptogram> contains an invalid sector public key BAD_ PK_{Sector} 7C <L_{7C}> 'A0' <L_{A0}> <BAD_PK_{Sector}> • <BAD_PK_{Sector}> is a sector public key which MUST differ from <PK_{Sector}>, i. e. hash(<BAD_PK_{Sector}>) MUST differ from the hash value encoded within terminal sector extension in AT_CERT_17c
Expected results	1. '90 00' within a valid Secure Messaging response 2. '63 00' or '6A 80'. The error MUST be encoded in a valid Secure Messaging response.

3.13.7 Test case EAC2_ISO7816_R_7

Test – ID	EAC2_ISO7816_R_7
Version	deleted in version 1.00 RC

3.13.8 Test case EAC2_ISO7816_R_8

Test – ID	EAC2_ISO7816_R_8
Purpose	Positive test for Restricted Identification, non-official Certificate
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_18, AT_CERT_18c) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '80' <L_{80}> <id-RI-x> '84' <L_{84}> <RefKeyID> 2. Send the given General Authenticate APDU to the eID Card: '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the sector public key PK_{Sector} 7C <L_{7C}> 'A0' <L_{A0}> <PK_{Sector}>, Hash(PK_{Sector}) MUST fit the hash value encoded in AT_CERT_18c

Expected results	1. ’90 00’ within a valid Secure Messaging response 2. 7C <L_{7C}> ‘81’ <L_{81}> <$I_{SectorPICC}$> ’90 00’ in valid Secure Messaging response

3.13.9 Test case EAC2_ISO7816_R_9

Test – ID	EAC2_ISO7816_R_9
Version	deleted in version 1.00 RC

3.13.10 Test case EAC2_ISO7816_R_10

Test – ID	EAC2_ISO7816_R_10
Purpose	Positive test for Restricted Identification, checking identical calculation of sector identifier
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_24 AT_CERT_24) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: ‘0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: ‘80’ <L_{80}> <id-RI-x> ‘84’ <L_{84}> <RefKeyID> 2. Send the given General Authenticate APDU to the eID Card: ‘0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>’ • <Cryptogram> contains the sector public key PK_{Sector} 7C <L_{7C}> ‘A0’ <L_{A0}> <PK_{Sector}>, Hash(PK_{Sector}) MUST fit the first hash value encoded in AT_CERT_24 3. Store returned <$I_{SectorPICC}$> 4. 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 AT APDU to the eID Card: ‘0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: ‘80’ <L_{80}> <id-RI-x> ‘84’ <L_{84}> <RefKeyID> 6. Send the given General Authenticate APDU to the eID Card: ‘0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>’

	• <Cryptogram> contains the sector public key PK_{Sector} 7C <L_{7C}> 'A0' <L_{A0}> <PK_{Sector}>, Hash(PK_{Sector}) MUST fit the first hash value encoded in AT_CERT_24 7. Stored <$I_{SectorPICC}$> MUST be identical to returned <$I_{SectorPICC}$>
Expected results	1. '90 00' within a valid Secure Messaging response 2. 7C <L_{7C}> '81' <L_{81}> <$I_{SectorPICC}$> '90 00' within a valid Secure Messaging response 3. true 4. true 5. '90 00' within a valid Secure Messaging response 6. 7C <L_{7C}> '81' <L_{81}> <$I_{SectorPICC}$> '90 00' within a valid Secure Messaging response 7. true

3.13.11 Test case EAC2_ISO7816_R_11

Test – ID	EAC2_ISO7816_R_11
Version	deleted in version 1.00 RC

3.13.12 Test case EAC2_ISO7816_R_12

Test – ID	EAC2_ISO7816_R_12
Purpose	Positive test for Restricted Identification, checking different calculation of sector identifier with different sector public keys and identical secret key, "migration scenario"
Version	EAC2_1.0
Profile	eID, RI
Preconditions	1. The PACE mechanism MUST have been performed, restricted identification must be allowed by CHAT 2. The Terminal Authentication MUST have been performed (DV_CERT_24 AT_CERT_24) 3. The Chip Authentication MUST have been performed 4. The eID application MUST have been selected 5. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the given MSE:Set AT APDU to the eID Card: '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '80' <L_{80}> <id-RI-x> '84' <L_{84}> <RefKeyID> 2. Send the given General Authenticate APDU to the eID Card: '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the sector public key $PK_{Sector1}$ 7C <L_{7C}> 'A0' <L_{A0}> <$PK_{Sector1}$>, Hash($PK_{Sector1}$) MUST fit the

	first hash value encoded in AT_CERT_24 3. Store returned <$I_{SectorPICC1}$> 4. Send the given MSE:Set AT APDU to the eID Card: '0C 22 41 A4 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects: '80' <L_{80}> <id-RI-x> '84' <L_{84}> <RefKeyID> 5. Send the given General Authenticate APDU to the eID Card: '0C 86 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 97 <L_{97}> <Ne> 8E 08 <Checksum> <Le>' • <Cryptogram> contains the sector public key $PK_{Sector2}$ 7C <L_{7C}> 'A2' <L_{A2}> <$PK_{Sector2}$>, Hash($PK_{Sector2}$) MUST fit the second hash value encoded in AT_CERT_24 6. Stored <$I_{SectorPICC1}$> MUST be different to returned <$I_{SectorPICC2}$>
Expected results	1. '90 00' within a valid Secure Messaging response 2. 7C <L_{7C}> '81' <L_{81}> <$I_{SectorPICC1}$> '90 00' within a valid Secure Messaging response 3. true 4. '90 00' within a valid Secure Messaging response 5. 7C <L_{7C}> '83' <L_{83}> <$I_{SectorPICC2}$> '90 00' within a valid Secure Messaging response 6. true

3.14 Unit test EAC2_ISO7816_T_Envelope mechanism

This unit covers all tests about the envelope mechanism that can be used as an alternative to Extended Length. During this mechanism the commands ENVELOPE and GET RESPONSE are used to transmit commands or data in separate chunks. This mechanism can be used if the terminal or chip do not support Extended Length.

3.14.1 Test case EAC2_ISO7816_T_1

Test – ID	EAC2_ISO7816_T_1
Purpose	Positive test of the ENVELOPE / GET RESPONSE mechanism where a chain of Envelope commands is sent to the ID card. This test case is based on ISO7816_J_1.
Version	EAC2_1.1
Profile	eID, TA2, ENV
Preconditions	1. The PACE mechanism MUST have been performed. 2. All envelope commands must be sent in plain, all enveloped APDUs are sent as valid SecureMessaging APDUs. 3. All enveloped response data MUST be SM protected.
Test scenario	1. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <L_c> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L83> <Certification Authority Reference> • The Certification Authority Reference MUST be used as returned by the PACE mechanism. 2. Send the following Envelope command without SM '10 C2 00 00 <L_c> <enveloped command>' with the following enveloped command (<enveloped command>) to the ID Card and a chunk size of 128 bytes: Send the appropriate DV-Certificate as specified in the "Certificate Set 1" chapter as DV_CERT_1. '0C 2A 00 BE <L_c> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <L_e>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> Dependent on the length of the Verify Certificate command above, the Envelope command must be sent several times. The last Envelope command must use the CLA byte '00'to signal the last element of the chain. 3. Send the following Get Response command to the ID card: '00 C0 00 00 00' 4. Send the given MSE: Set DST APDU to the eID Card. '0C 22 81 B6 <L_c> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the following encrypted data objects 83 <L_{83}> <Certification Authority Reference> • The Certificate Holder Reference stored inside the DV-Certificate sent in step 2 has to be used. 5. Send the following Envelope command without SM '10 C2 00 00 <L_c> <enveloped command>' with the following enveloped command (<enveloped command>) to the ID Card and a chunk size of 128 bytes similar to step 2:

	Send the appropriate IS-Certificate as specified in the "Certificate Set 1" chapter as IS_CERT_1. '0C 2A 00 BE <L_c> 87 <L_{87}> 01 <Cryptogram> 8E 08 <Checksum> <L_e>' • <Cryptogram> contains the following encrypted data objects 7F 4E <L_{7F4E}> <certificate body> 5F 37 <L_{5F37}> <certificate signature> Dependent on the length of the Verify Certificate command above, the Envelope command must be sent several times. The last Envelope command must use the CLA byte '00'to signal the last element of the chain. 6. Send the following Get Response command to the eID Card: '00 C0 00 00 00'
Expected results	1. '90 00' within a valid Secure Messaging response 2. All Envelope commands must return '90 00' in plain, except the last Envelope command must return '61 xx' 3. Get Response command must return '90 00' in plain The enveloped Verify Certificate command must return '90 00' within a valid Secure Messaging response 4. '90 00' within a valid Secure Messaging response 5. All Envelope commands must return '90 00' in plain, except the last Envelope command must return '61 xx' 6. Get Response command must return '90 00' in plain The enveloped Verify Certificate command must return '90 00' within a valid Secure Messaging response

3.14.2 Test case EAC2_ISO7816_T_2

Test – ID	EAC2_ISO7816_T_2
Purpose	Positive test of the ENVELOPE / GET RESPONSE mechanism where a chain of Get Response commands is sent to the ID card.
Version	EAC2_1.1
Profile	eID, ENV, TA2, CA2, PACE, ePassport
Preconditions	1. The PACE mechanism MUST have been performed 2. The Terminal Authentication MUST have been performed 3. The Chip Authentication MUST have been performed 4. The eMRTD application MUST have been selected 5. All enveloped APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the following Envelope command to the eID Card with an enveloped Read Binary command to read the first six bytes of EF.DG2: '00 C2 00 00 <L_c> 0C B0 82 00 0D 97 01 06 8E 08 <checksum> 00' 2. Send the following Get Response command to the eID Card:

	‘00 C0 00 00 00’
Expected results	1. ’61 xx’ in plain where xx is the number of remaining bytes 2. Get Response must return ‘90 00’ Enveloped Read Binary command must return ‘90 00’within a valid Secure Messaging response At the end the returned data must include the first six bytes of DG2 and be valid.

3.14.3 Test case EAC2_ISO7816_T_3

Test – ID	EAC2_ISO7816_T_3
Purpose	Negative test of the ENVELOPE / GET RESPONSE mechanism to test that the chip reacts correct if a Get Response command is performed without a previous Envelope command.
Version	EAC2_1.1
Profile	eID, ENV
Preconditions	1. None, card recently activated
Test scenario	1. Send the given Get Response APDU to the eID Card: ‘00 C0 00 00 00’
Expected results	1. Checking error; the error code SHALL be returned as plain data without SM encoding and without any data

3.14.4 Test case EAC2_ISO7816_T_4

Test – ID	EAC2_ISO7816_T_4
Purpose	Negative test of the ENVELOPE / GET RESPONSE mechanism to test that the chip reacts correct if ENVELOPE / GET RESPONSE mechanism is performed in an SM channel.
Version	EAC2_1.1
Profile	eID, ENV
Preconditions	1. The PACE mechanism MUST have been performed 2. All APDUs are sent as valid secure messaging APDUs
Test scenario	1. Send the following Envelope command without SM ‘00 C2 00 00 <L_c> <enveloped command>’ with the following enveloped command (<enveloped command>) to the ID Card: ‘0C B0 (80 | <sfi.EF.CardAccess>) 00 0D 97 01 01 8E 08 <checksum> 00’ 2. Send the given Get Response APDU to the eID Card: ‘0C C0 00 00 <L_c> 97 01 00 8E 08 <Checksum> 00’
Expected results	1. ’61 xx’ in plain 2. Checking error in an SM encoded response APDU

3.14.5 Test case EAC2_ISO7816_T_5

Test – ID	EAC2_ISO7816_T_5
Purpose	Negative test of the ENVELOPE / GET RESPONSE mechanism to test that chip can return different status words – on the one hand a status word for the Envelope command and on the other hand a different status word for the enveloped command itself.
Version	EAC2_1.1
Profile	eID, ENV
Preconditions	1. None, card recently activated
Test scenario	1. Send the given Envelope APDU to the eID Card with an enveloped Read Binary (EF.CardSecurity) APDU: '00 C2 00 00 05 00 B0 9D 00 00' 2. Send the given Get Response APDU to the eID Card: '00 C0 00 00 00'
Expected results	1. '61 02' in plain 2. Get Response command: '90 00' in plain Enveloped status word: Checking error; the error code SHALL be returned as plain data without SM encoding and without any data

3.15 Unit test EAC2_ISO7816_U_Compare

This unit covers all tests about the command COMPARE that can be used to compare data groups in the eID application of the chip.

3.15.1 Test case EAC2_ISO7816_U_1_Template

Test – ID	EAC2_ISO7816_U_1
Purpose	Positive test of the Compare command. An official domestic certificate is used and the verification is successful.
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 18
Preconditions	1. The PACE mechanism MUST have been performed using PIN, verification according to table 18 must be allowed by Authorization Extension bits in tag 65 in command MSE:Set AT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT as defined in table 18 3. Compare data with valid data object as defined in table 18 MUST have been sent by authorized terminal during Terminal Authentication mechanism. Sent data MUST match the content of tested data group 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 18
Expected results	1. '90 00' within a valid Secure Messaging response.

Test case EAC2_ISO7816_U_1a to test case EAC2_ISO7816_U_1v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_1a	EAC2_1.1	Compare DG1	AT_CERT_30a	id-DGContent-DG1
EAC2_ISO7816_U_1b	EAC2_1.1	Compare DG2	AT_CERT_30b	id-DGContent-DG2
EAC2_ISO7816_U_1c	EAC2_1.1	Compare DG3	AT_CERT_30c	id-DGContent-DG3
EAC2_ISO7816_U_1d	EAC2_1.1	Compare DG4	AT_CERT_30d	id-DGContent-DG4
EAC2_ISO7816_U_1e	EAC2_1.1	Compare DG5	AT_CERT_30e	id-DGContent-DG5
EAC2_ISO7816_U_1f	EAC2_1.1	Compare DG6	AT_CERT_30f	id-DGContent-DG6
EAC2_ISO7816_U_1g	EAC2_1.1	Compare DG7	AT_CERT_30g	id-DGContent-DG7

EAC2_ISO7816_U_1h	EAC2_1.1	Compare DG8	AT_CERT_30h	id-DGContent-DG8
EAC2_ISO7816_U_1i	EAC2_1.1	Compare DG9	AT_CERT_30i	id-DGContent-DG9
EAC2_ISO7816_U_1j	EAC2_1.1	Compare DG10	AT_CERT_30j	id-DGContent-DG10
EAC2_ISO7816_U_1k	EAC2_1.1	Compare DG11	AT_CERT_30k	id-DGContent-DG11
EAC2_ISO7816_U_1l	EAC2_1.1	Compare DG12	AT_CERT_30l	id-DGContent-DG12
EAC2_ISO7816_U_1m	EAC2_1.1	Compare DG13	AT_CERT_30m	id-DGContent-DG13
EAC2_ISO7816_U_1n	EAC2_1.1	Compare DG14	AT_CERT_30n	id-DGContent-DG14
EAC2_ISO7816_U_1o	EAC2_1.1	Compare DG15	AT_CERT_30o	id-DGContent-DG15
EAC2_ISO7816_U_1p	EAC2_1.1	Compare DG16	AT_CERT_30p	id-DGContent-DG16
EAC2_ISO7816_U_1q	EAC2_1.1	Compare DG17	AT_CERT_30q	id-DGContent-DG17
EAC2_ISO7816_U_1r	EAC2_1.1	Compare DG18	AT_CERT_30r	id-DGContent-DG18
EAC2_ISO7816_U_1s	EAC2_1.1	Compare DG19	AT_CERT_30s	id-DGContent-DG19
EAC2_ISO7816_U_1t	EAC2_1.1	Compare DG20	AT_CERT_30t	id-DGContent-DG20
EAC2_ISO7816_U_1u	EAC2_1.1	Compare DG21	AT_CERT_30u	id-DGContent-DG21
EAC2_ISO7816_U_1v	EAC2_1.1	Compare DG22	AT_CERT_30v	id-DGContent-DG22

Table 18: Test cases EAC2_ISO7816_U_1

3.15.2 Test case EAC2_ISO7816_U_2_Template

Test – ID	EAC2_ISO7816_U_2
Purpose	Negative test of the Compare command. An official domestic certificate is used and the verification fails.
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 19
Preconditions	1. The PACE mechanism MUST have been performed using PIN, verification according to table 19 must be allowed by Authorization Extension bits in tag 65 in command MSE: Set AT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in “Certificate Set 30“ chapter as DV_CERT_30 and AT_CERT according to table 19 3. During Terminal Authentication data to be compared MUST be sent with a valid data object as defined in table 19. The sent content MUST NOT match the content stored in corresponding data group 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. ‘0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08

	<Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 19
Expected results	1. '63 40'. The error MUST be encoded in a valid Secure Messaging response.

Test case EAC2_ISO7816_U_2a to test case EAC2_ISO7816_U_2v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_2a	EAC2_1.1	Compare DG1	AT_CERT_30a	id-DGContent-DG1
EAC2_ISO7816_U_2b	EAC2_1.1	Compare DG2	AT_CERT_30b	id-DGContent-DG2
EAC2_ISO7816_U_2c	EAC2_1.1	Compare DG3	AT_CERT_30c	id-DGContent-DG3
EAC2_ISO7816_U_2d	EAC2_1.1	Compare DG4	AT_CERT_30d	id-DGContent-DG4
EAC2_ISO7816_U_2e	EAC2_1.1	Compare DG5	AT_CERT_30e	id-DGContent-DG5
EAC2_ISO7816_U_2f	EAC2_1.1	Compare DG6	AT_CERT_30f	id-DGContent-DG6
EAC2_ISO7816_U_2g	EAC2_1.1	Compare DG7	AT_CERT_30g	id-DGContent-DG7
EAC2_ISO7816_U_2h	EAC2_1.1	Compare DG8	AT_CERT_30h	id-DGContent-DG8
EAC2_ISO7816_U_2i	EAC2_1.1	Compare DG9	AT_CERT_30i	id-DGContent-DG9
EAC2_ISO7816_U_2j	EAC2_1.1	Compare DG10	AT_CERT_30j	id-DGContent-DG10
EAC2_ISO7816_U_2k	EAC2_1.1	Compare DG11	AT_CERT_30k	id-DGContent-DG11
EAC2_ISO7816_U_2l	EAC2_1.1	Compare DG12	AT_CERT_30l	id-DGContent-DG12
EAC2_ISO7816_U_2m	EAC2_1.1	Compare DG13	AT_CERT_30m	id-DGContent-DG13
EAC2_ISO7816_U_2n	EAC2_1.1	Compare DG14	AT_CERT_30n	id-DGContent-DG14
EAC2_ISO7816_U_2o	EAC2_1.1	Compare DG15	AT_CERT_30o	id-DGContent-DG15
EAC2_ISO7816_U_2p	EAC2_1.1	Compare DG16	AT_CERT_30p	id-DGContent-DG16
EAC2_ISO7816_U_2q	EAC2_1.1	Compare DG17	AT_CERT_30q	id-DGContent-DG17
EAC2_ISO7816_U_2r	EAC2_1.1	Compare DG18	AT_CERT_30r	id-DGContent-DG18
EAC2_ISO7816_U_2s	EAC2_1.1	Compare DG19	AT_CERT_30s	id-DGContent-DG19
EAC2_ISO7816_U_2t	EAC2_1.1	Compare DG20	AT_CERT_30t	id-DGContent-DG20
EAC2_ISO7816_U_2u	EAC2_1.1	Compare DG21	AT_CERT_30u	id-DGContent-DG21
EAC2_ISO7816_U_2v	EAC2_1.1	Compare DG22	AT_CERT_30v	id-DGContent-DG22

Table 19: Test cases EAC2_ISO7816_U_2

3.15.3 Test case EAC2_ISO7816_U_3_Template

Test – ID	EAC2_ISO7816_U_3
Purpose	Negative test of Compare, verification fails because of unauthorized certificate, official domestic certificate
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 20
Preconditions	1. The PACE mechanism MUST have been performed using PIN, verification according to table 20 must be allowed by Authorization Extension bits in tag 65 in command MSE:Set AT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT_30w as defined in table 20 3. Compare data with valid data object as defined in table 20 MUST have been sent by an authorized terminal during the Terminal Authentication mechanism. Sent data MUST match the content of tested data group 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 20
Expected results	1. '69 82'. The error MUST be encoded in a valid Secure Messaging response.

Test case EAC2_ISO7816_U_3a to test case EAC2_ISO7816_U_3v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_3a	EAC2_1.1	Compare DG1	AT_CERT_30w	id-DGContent-DG1
EAC2_ISO7816_U_3b	EAC2_1.1	Compare DG2	AT_CERT_30w	id-DGContent-DG2
EAC2_ISO7816_U_3c	EAC2_1.1	Compare DG3	AT_CERT_30w	id-DGContent-DG3
EAC2_ISO7816_U_3d	EAC2_1.1	Compare DG4	AT_CERT_30w	id-DGContent-DG4
EAC2_ISO7816_U_3e	EAC2_1.1	Compare DG5	AT_CERT_30w	id-DGContent-DG5
EAC2_ISO7816_U_3f	EAC2_1.1	Compare DG6	AT_CERT_30w	id-DGContent-DG6
EAC2_ISO7816_U_3g	EAC2_1.1	Compare DG7	AT_CERT_30w	id-DGContent-DG7
EAC2_ISO7816_U_3h	EAC2_1.1	Compare DG8	AT_CERT_30w	id-DGContent-DG8
EAC2_ISO7816_U_3i	EAC2_1.1	Compare DG9	AT_CERT_30w	id-DGContent-DG9
EAC2_ISO7816_U_3j	EAC2_1.1	Compare DG10	AT_CERT_30w	id-DGContent-DG10

EAC2_ISO7816_U_3k	EAC2_1.1	Compare DG11	AT_CERT_30w	id-DGContent-DG11
EAC2_ISO7816_U_3l	EAC2_1.1	Compare DG12	AT_CERT_30w	id-DGContent-DG12
EAC2_ISO7816_U_3m	EAC2_1.1	Compare DG13	AT_CERT_30w	id-DGContent-DG13
EAC2_ISO7816_U_3n	EAC2_1.1	Compare DG14	AT_CERT_30w	id-DGContent-DG14
EAC2_ISO7816_U_3o	EAC2_1.1	Compare DG15	AT_CERT_30w	id-DGContent-DG15
EAC2_ISO7816_U_3p	EAC2_1.1	Compare DG16	AT_CERT_30w	id-DGContent-DG16
EAC2_ISO7816_U_3q	EAC2_1.1	Compare DG17	AT_CERT_30w	id-DGContent-DG17
EAC2_ISO7816_U_3r	EAC2_1.1	Compare DG18	AT_CERT_30w	id-DGContent-DG18
EAC2_ISO7816_U_3s	EAC2_1.1	Compare DG19	AT_CERT_30w	id-DGContent-DG19
EAC2_ISO7816_U_3t	EAC2_1.1	Compare DG20	AT_CERT_30w	id-DGContent-DG20
EAC2_ISO7816_U_3u	EAC2_1.1	Compare DG21	AT_CERT_30w	id-DGContent-DG21
EAC2_ISO7816_U_3v	EAC2_1.1	Compare DG22	AT_CERT_30w	id-DGContent-DG22

Table 20: Test cases EAC2_ISO7816_U_3

3.15.4 Test case EAC2_ISO7816_U_4_Template

Test – ID	EAC2_ISO7816_U_4
Purpose	Negative test of the Compare command. The verification fails because no data for comparison has been transmitted. An official domestic certificate is used.
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 21
Preconditions	1. The PACE mechanism MUST have been performed using PIN, verification as in table 21 must be allowed by Authorization Extension bits in tag 65 in command MSE:Set AT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in the "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT defined according to table 21 3. Compare data with valid data object as defined in table 21 MUST have been sent by authorized terminal during Terminal Authentication mechanism. Data to be compared MUST NOT be sent using a valid AAD including an DDT with empty discretionary data and valid OID 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 21

Expected results	1. ’6340’. The error MUST be encoded in a valid Secure Messaging response.

Test case EAC2_ISO7816_U_4a to test case EAC2_ISO7816_U_4v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_4a	EAC2_1.1	Compare DG1	AT_CERT_30a	id-DGContent-DG1
EAC2_ISO7816_U_4b	EAC2_1.1	Compare DG2	AT_CERT_30b	id-DGContent-DG2
EAC2_ISO7816_U_4c	EAC2_1.1	Compare DG3	AT_CERT_30c	id-DGContent-DG3
EAC2_ISO7816_U_4d	EAC2_1.1	Compare DG4	AT_CERT_30d	id-DGContent-DG4
EAC2_ISO7816_U_4e	EAC2_1.1	Compare DG5	AT_CERT_30e	id-DGContent-DG5
EAC2_ISO7816_U_4f	EAC2_1.1	Compare DG6	AT_CERT_30f	id-DGContent-DG6
EAC2_ISO7816_U_4g	EAC2_1.1	Compare DG7	AT_CERT_30g	id-DGContent-DG7
EAC2_ISO7816_U_4h	EAC2_1.1	Compare DG8	AT_CERT_30h	id-DGContent-DG8
EAC2_ISO7816_U_4i	EAC2_1.1	Compare DG9	AT_CERT_30i	id-DGContent-DG9
EAC2_ISO7816_U_4j	EAC2_1.1	Compare DG10	AT_CERT_30j	id-DGContent-DG10
EAC2_ISO7816_U_4k	EAC2_1.1	Compare DG11	AT_CERT_30k	id-DGContent-DG11
EAC2_ISO7816_U_4l	EAC2_1.1	Compare DG12	AT_CERT_30l	id-DGContent-DG12
EAC2_ISO7816_U_4m	EAC2_1.1	Compare DG13	AT_CERT_30m	id-DGContent-DG13
EAC2_ISO7816_U_4n	EAC2_1.1	Compare DG14	AT_CERT_30n	id-DGContent-DG14
EAC2_ISO7816_U_4o	EAC2_1.1	Compare DG15	AT_CERT_30o	id-DGContent-DG15
EAC2_ISO7816_U_4p	EAC2_1.1	Compare DG16	AT_CERT_30p	id-DGContent-DG16
EAC2_ISO7816_U_4q	EAC2_1.1	Compare DG17	AT_CERT_30q	id-DGContent-DG17
EAC2_ISO7816_U_4r	EAC2_1.1	Compare DG18	AT_CERT_30r	id-DGContent-DG18
EAC2_ISO7816_U_4s	EAC2_1.1	Compare DG19	AT_CERT_30s	id-DGContent-DG19
EAC2_ISO7816_U_4t	EAC2_1.1	Compare DG20	AT_CERT_30t	id-DGContent-DG20
EAC2_ISO7816_U_4u	EAC2_1.1	Compare DG21	AT_CERT_30u	id-DGContent-DG21
EAC2_ISO7816_U_4v	EAC2_1.1	Compare DG22	AT_CERT_30v	id-DGContent-DG22

Table 21: Test cases EAC2_ISO7816_U_4

3.15.5 Test case EAC2_ISO7816_U_5_Template

Test – ID	EAC2_ISO7816_U_5
Purpose	Positive test of Compare, verification successful, non-official certificate

Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 22
Preconditions	1. The PACE mechanism MUST have been performed using PIN, verification as in table 22 must be allowed by Authorization Extension bits in tag 65 in command MSE:Set AT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in "Certificate Set 31" chapter as DV_CERT_31 and AT_CERT as defined in table 22 3. Compare data with valid data object as defined in table 22 MUST have been sent by authorized terminal during Terminal Authentication mechanism. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 22
Expected results	1. '90 00' within a valid Secure Messaging response.

Test case EAC2_ISO7816_U_5a to test case EAC2_ISO7816_U_5v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_5a	EAC2_1.1	Compare DG1	AT_CERT_31a	id-DGContent-DG1
EAC2_ISO7816_U_5b	EAC2_1.1	Compare DG2	AT_CERT_31b	id-DGContent-DG2
EAC2_ISO7816_U_5c	EAC2_1.1	Compare DG3	AT_CERT_31c	id-DGContent-DG3
EAC2_ISO7816_U_5d	EAC2_1.1	Compare DG4	AT_CERT_31d	id-DGContent-DG4
EAC2_ISO7816_U_5e	EAC2_1.1	Compare DG5	AT_CERT_31e	id-DGContent-DG5
EAC2_ISO7816_U_5f	EAC2_1.1	Compare DG6	AT_CERT_31f	id-DGContent-DG6
EAC2_ISO7816_U_5g	EAC2_1.1	Compare DG7	AT_CERT_31g	id-DGContent-DG7
EAC2_ISO7816_U_5h	EAC2_1.1	Compare DG8	AT_CERT_31h	id-DGContent-DG8
EAC2_ISO7816_U_5i	EAC2_1.1	Compare DG9	AT_CERT_31i	id-DGContent-DG9
EAC2_ISO7816_U_5j	EAC2_1.1	Compare DG10	AT_CERT_31j	id-DGContent-DG10
EAC2_ISO7816_U_5k	EAC2_1.1	Compare DG11	AT_CERT_31k	id-DGContent-DG11
EAC2_ISO7816_U_5l	EAC2_1.1	Compare DG12	AT_CERT_31l	id-DGContent-DG12
EAC2_ISO7816_U_5m	EAC2_1.1	Compare DG13	AT_CERT_31m	id-DGContent-DG13
EAC2_ISO7816_U_5n	EAC2_1.1	Compare DG14	AT_CERT_31n	id-DGContent-DG14

EAC2_ISO7816_U_5o	EAC2_1.1	Compare DG15	AT_CERT_31o	id-DGContent-DG15
EAC2_ISO7816_U_5p	EAC2_1.1	Compare DG16	AT_CERT_31p	id-DGContent-DG16
EAC2_ISO7816_U_5q	EAC2_1.1	Compare DG17	AT_CERT_31q	id-DGContent-DG17
EAC2_ISO7816_U_5r	EAC2_1.1	Compare DG18	AT_CERT_31r	id-DGContent-DG18
EAC2_ISO7816_U_5s	EAC2_1.1	Compare DG19	AT_CERT_31s	id-DGContent-DG19
EAC2_ISO7816_U_5t	EAC2_1.1	Compare DG20	AT_CERT_31t	id-DGContent-DG20
EAC2_ISO7816_U_5u	EAC2_1.1	Compare DG21	AT_CERT_31u	id-DGContent-DG21
EAC2_ISO7816_U_5v	EAC2_1.1	Compare DG22	AT_CERT_31v	id-DGContent-DG22

Table 22: Test cases EAC2_ISO7816_U_5

3.15.6 Test case EAC2_ISO7816_U_6_Template

Test – ID	EAC2_ISO7816_U_6
Purpose	Negative test of Compare, verification fail because of missing tag 65 for Authentication Extension in MSE:Set AT during PACE
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 23
Preconditions	1. The PACE mechanism MUST have been performed using PIN, but without tag 65 in command MSE:Set AT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in “Certificate Set 30“ chapter as DV_CERT_30 and AT_CERT as defined in table 23 3. Compare data with valid data object as defined in table 23 MUST have been sent by authorized terminal during Terminal Authentication mechanism. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. ‘0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the encrypted ‘data object’ as defined in table 23
Expected results	1. Checking error in valid Secure Messaging response.

Test case EAC2_ISO7816_U_6a to test case EAC2_ISO7816_U_6v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_6a	EAC2_1.1	Compare DG1	AT_CERT_30a	id-DGContent-DG1
EAC2_ISO7816_U_6b	EAC2_1.1	Compare DG2	AT_CERT_30b	id-DGContent-DG2

EAC2_ISO7816_U_6c	EAC2_1.1	Compare DG3	AT_CERT_30c	id-DGContent-DG3
EAC2_ISO7816_U_6d	EAC2_1.1	Compare DG4	AT_CERT_30d	id-DGContent-DG4
EAC2_ISO7816_U_6e	EAC2_1.1	Compare DG5	AT_CERT_30e	id-DGContent-DG5
EAC2_ISO7816_U_6f	EAC2_1.1	Compare DG6	AT_CERT_30f	id-DGContent-DG6
EAC2_ISO7816_U_6g	EAC2_1.1	Compare DG7	AT_CERT_30g	id-DGContent-DG7
EAC2_ISO7816_U_6h	EAC2_1.1	Compare DG8	AT_CERT_30h	id-DGContent-DG8
EAC2_ISO7816_U_6i	EAC2_1.1	Compare DG9	AT_CERT_30i	id-DGContent-DG9
EAC2_ISO7816_U_6j	EAC2_1.1	Compare DG10	AT_CERT_30j	id-DGContent-DG10
EAC2_ISO7816_U_6k	EAC2_1.1	Compare DG11	AT_CERT_30k	id-DGContent-DG11
EAC2_ISO7816_U_6l	EAC2_1.1	Compare DG12	AT_CERT_30l	id-DGContent-DG12
EAC2_ISO7816_U_6m	EAC2_1.1	Compare DG13	AT_CERT_30m	id-DGContent-DG13
EAC2_ISO7816_U_6n	EAC2_1.1	Compare DG14	AT_CERT_30n	id-DGContent-DG14
EAC2_ISO7816_U_6o	EAC2_1.1	Compare DG15	AT_CERT_30o	id-DGContent-DG15
EAC2_ISO7816_U_6p	EAC2_1.1	Compare DG16	AT_CERT_30p	id-DGContent-DG16
EAC2_ISO7816_U_6q	EAC2_1.1	Compare DG17	AT_CERT_30q	id-DGContent-DG17
EAC2_ISO7816_U_6r	EAC2_1.1	Compare DG18	AT_CERT_30r	id-DGContent-DG18
EAC2_ISO7816_U_6s	EAC2_1.1	Compare DG19	AT_CERT_30s	id-DGContent-DG19
EAC2_ISO7816_U_6t	EAC2_1.1	Compare DG20	AT_CERT_30t	id-DGContent-DG20
EAC2_ISO7816_U_6u	EAC2_1.1	Compare DG21	AT_CERT_30u	id-DGContent-DG21
EAC2_ISO7816_U_6v	EAC2_1.1	Compare DG22	AT_CERT_30v	id-DGContent-DG22

Table 23: Test cases EAC2_ISO7816_U_6

3.15.7 Test case EAC2_ISO7816_U_7_Template

Test – ID	EAC2_ISO7816_U_7
Purpose	Negative test of the Compare mechanism. The verification fails because of a mismatch between the access rights used during PACE encoded in the Authorization Extensions bits sent by the MSE:Set AT command and the access rights of the certificate used during TA.
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 24
Preconditions	1. The PACE mechanism MUST have been performed using PIN, verification according to table 24 must not be allowed by Authorization Extension bits in tag 65. Use Authorization Extensions without any rights. 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in “Certificate Set 30“ chapter as DV_CERT_30

	and AT_CERT as defined in table 24 3. Compare data with valid data object as defined in table 24 MUST have been sent by authorized terminal during Terminal Authentication mechanism. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 24
Expected results	1. Checking error in valid Secure Messaging response.

Test case EAC2_ISO7816_U_7a to test case EAC2_ISO7816_U_7v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_7a	EAC2_1.1	Compare DG1	AT_CERT_30a	id-DGContent-DG1
EAC2_ISO7816_U_7b	EAC2_1.1	Compare DG2	AT_CERT_30b	id-DGContent-DG2
EAC2_ISO7816_U_7c	EAC2_1.1	Compare DG3	AT_CERT_30c	id-DGContent-DG3
EAC2_ISO7816_U_7d	EAC2_1.1	Compare DG4	AT_CERT_30d	id-DGContent-DG4
EAC2_ISO7816_U_7e	EAC2_1.1	Compare DG5	AT_CERT_30e	id-DGContent-DG5
EAC2_ISO7816_U_7f	EAC2_1.1	Compare DG6	AT_CERT_30f	id-DGContent-DG6
EAC2_ISO7816_U_7g	EAC2_1.1	Compare DG7	AT_CERT_30g	id-DGContent-DG7
EAC2_ISO7816_U_7h	EAC2_1.1	Compare DG8	AT_CERT_30h	id-DGContent-DG8
EAC2_ISO7816_U_7i	EAC2_1.1	Compare DG9	AT_CERT_30i	id-DGContent-DG9
EAC2_ISO7816_U_7j	EAC2_1.1	Compare DG10	AT_CERT_30j	id-DGContent-DG10
EAC2_ISO7816_U_7k	EAC2_1.1	Compare DG11	AT_CERT_30k	id-DGContent-DG11
EAC2_ISO7816_U_7l	EAC2_1.1	Compare DG12	AT_CERT_30l	id-DGContent-DG12
EAC2_ISO7816_U_7m	EAC2_1.1	Compare DG13	AT_CERT_30m	id-DGContent-DG13
EAC2_ISO7816_U_7n	EAC2_1.1	Compare DG14	AT_CERT_30n	id-DGContent-DG14
EAC2_ISO7816_U_7o	EAC2_1.1	Compare DG15	AT_CERT_30o	id-DGContent-DG15
EAC2_ISO7816_U_7p	EAC2_1.1	Compare DG16	AT_CERT_30p	id-DGContent-DG16
EAC2_ISO7816_U_7q	EAC2_1.1	Compare DG17	AT_CERT_30q	id-DGContent-DG17
EAC2_ISO7816_U_7r	EAC2_1.1	Compare DG18	AT_CERT_30r	id-DGContent-DG18
EAC2_ISO7816_U_7s	EAC2_1.1	Compare DG19	AT_CERT_30s	id-DGContent-DG19

EAC2_ISO7816_U_7t	EAC2_1.1	Compare DG20	AT_CERT_30t	id-DGContent-DG20
EAC2_ISO7816_U_7u	EAC2_1.1	Compare DG21	AT_CERT_30u	id-DGContent-DG21
EAC2_ISO7816_U_7v	EAC2_1.1	Compare DG22	AT_CERT_30v	id-DGContent-DG22

Table 24: Test cases EAC2_ISO7816_U_7

3.15.8 Test case EAC2_ISO7816_U_8_Template

Test – ID	EAC2_ISO7816_U_8
Purpose	Negative test of Compare, verification fail because of empty tag 65 for Authentication Extension in MSE:Set AT during PACE
Version	EAC2_1.1
Profile	eID, CMP, AUTH_EXT, required data group presence see table 25
Preconditions	1. The PACE mechanism MUST have been performed using PIN and MSE:Set AT MUST contain an empty tag 65 without any data 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT as defined in table 25 3. Compare data with valid data object as defined in table 25 MUST have been sent by authorized terminal during Terminal Authentication mechanism. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <L_c> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' • <Cryptogram> contains the encrypted 'data object' as defined in table 25
Expected results	1. Checking error in valid Secure Messaging response.

Test case EAC2_ISO7816_U_8a to test case EAC2_ISO7816_U_8v:

Testcase-ID	**Version**	**Access Rules**	**Cert Reference**	**Data Object**
EAC2_ISO7816_U_8a	EAC2_1.1	Compare DG1	AT_CERT_30a	id-DGContent-DG1
EAC2_ISO7816_U_8b	EAC2_1.1	Compare DG2	AT_CERT_30b	id-DGContent-DG2
EAC2_ISO7816_U_8c	EAC2_1.1	Compare DG3	AT_CERT_30c	id-DGContent-DG3
EAC2_ISO7816_U_8d	EAC2_1.1	Compare DG4	AT_CERT_30d	id-DGContent-DG4
EAC2_ISO7816_U_8e	EAC2_1.1	Compare DG5	AT_CERT_30e	id-DGContent-DG5
EAC2_ISO7816_U_8f	EAC2_1.1	Compare DG6	AT_CERT_30f	id-DGContent-DG6
EAC2_ISO7816_U_8g	EAC2_1.1	Compare DG7	AT_CERT_30g	id-DGContent-DG7

EAC2_ISO7816_U_8h	EAC2_1.1	Compare DG8	AT_CERT_30h	id-DGContent-DG8
EAC2_ISO7816_U_8i	EAC2_1.1	Compare DG9	AT_CERT_30i	id-DGContent-DG9
EAC2_ISO7816_U_8j	EAC2_1.1	Compare DG10	AT_CERT_30j	id-DGContent-DG10
EAC2_ISO7816_U_8k	EAC2_1.1	Compare DG11	AT_CERT_30k	id-DGContent-DG11
EAC2_ISO7816_U_8l	EAC2_1.1	Compare DG12	AT_CERT_30l	id-DGContent-DG12
EAC2_ISO7816_U_8m	EAC2_1.1	Compare DG13	AT_CERT_30m	id-DGContent-DG13
EAC2_ISO7816_U_8n	EAC2_1.1	Compare DG14	AT_CERT_30n	id-DGContent-DG14
EAC2_ISO7816_U_8o	EAC2_1.1	Compare DG15	AT_CERT_30o	id-DGContent-DG15
EAC2_ISO7816_U_8p	EAC2_1.1	Compare DG16	AT_CERT_30p	id-DGContent-DG16
EAC2_ISO7816_U_8q	EAC2_1.1	Compare DG17	AT_CERT_30q	id-DGContent-DG17
EAC2_ISO7816_U_8r	EAC2_1.1	Compare DG18	AT_CERT_30r	id-DGContent-DG18
EAC2_ISO7816_U_8s	EAC2_1.1	Compare DG19	AT_CERT_30s	id-DGContent-DG19
EAC2_ISO7816_U_8t	EAC2_1.1	Compare DG20	AT_CERT_30t	id-DGContent-DG20
EAC2_ISO7816_U_8u	EAC2_1.1	Compare DG21	AT_CERT_30u	id-DGContent-DG21
EAC2_ISO7816_U_8v	EAC2_1.1	Compare DG22	AT_CERT_30v	id-DGContent-DG22

Table 25: Test cases EAC2_ISO7816_U_8

3.15.9 Test case EAC2_ISO7816_U_9

Test – ID	EAC2_ISO7816_U_9
Purpose	Positive test of Compare command with two Discretionary Data Templates in Authentication Data Object containing age verification data and document validity verification data, verification successful, official domestic certificate
Version	EAC2_1.1
Profile	eID, CMP
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in “Certificate Set 30“ chapter as DV_CERT_30 and AT_CERT_30x. 3. Auxiliary data with valid Date of Birth data object and document validity data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Date of birth and document validity MUST fit the required age and validity. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card.

	‘0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth> 2. Send the given Compare APDU to the eID Card. ‘0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response.

3.15.10 Test case EAC2_ISO7816_U_10

Test – ID	EAC2_ISO7816_U_10
Purpose	Positive test with three Discretionary Data Templates in Authentication Data Object containing age verification data, document validity verification data and MunicipalityID verification data, verification successful, official domestic certificate
Version	EAC2_1.1
Profile	eID, CMP
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in “Certificate Set 30“ chapter as DV_CERT_30 and AT_CERT_30x. 3. Auxiliary data with valid Date of Birth data object, document validity data object and MunicipalityID data object MUST have been sent by authorized terminal during Terminal Authentication mechanism. Date of birth, document validity and MunicipalityID MUST fit the required age, validity and MunicipalityID. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. ‘0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth> 2. Send the given Compare APDU to the eID Card. ‘0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfExpiry> 3. Send the given Compare APDU to the eID Card. ‘ 0C 33 00 00 <Lc> 87 <L_{87}> 01 <Cryptogram> 8E 08

	<Checksum> 00’ • <Cryptogram> contains the following encrypted data objects: <id-MunicipalityID>
Expected results	1. ’90 00’ within a valid Secure Messaging response. 2. ’90 00’ within a valid Secure Messaging response. 3. ’90 00’ within a valid Secure Messaging response.

3.15.11 Test case EAC2_ISO7816_U_11

Test – ID	EAC2_ISO7816_U_11
Purpose	Positive test with two Discretionary Data Templates of the same type in Authentication Data Object containing two age verification data objects with same OID, verification of last data object successful, official domestic certificate
Version	EAC2_1.1
Profile	eID, CMP
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in “Certificate Set 30“ chapter as DV_CERT_30 and AT_CERT_30x. 3. Auxiliary data with two valid Date of Birth data objects MUST have been sent by authorized terminal during Terminal Authentication mechanism. The first date of birth MUST NOT fit the required age, the second date of birth MUST fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. ‘0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00’ <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. ’90 00’ within a valid Secure Messaging response.

3.15.12 Test case EAC2_ISO7816_U_12

Test – ID	EAC2_ISO7816_U_12
Purpose	Negative test with two Discretionary Data Templates of the same type in Authentication Data Object containing two age verification data objects with same OID, verification of last data object fail, official domestic certificate
Version	EAC2_1.1
Profile	eID, CMP
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with

	the certificates defined in "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT_30x. 3. Auxiliary data with two valid Date of Birth data objects MUST have been sent by authorized terminal during Terminal Authentication mechanism. The first date of birth MUST fit the required age, the second date of birth MUST NOT fit the required age. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. '63 40' or Checking error in a valid Secure Messaging response.

3.15.13 Test case EAC2_ISO7816_U_13

Test – ID	EAC2_ISO7816_U_13
Purpose	Negative test with invalid Authentication Data Object containing an incorrect Discretionary Data Template tag
Version	EAC2_1.1
Profile	eID, CMP
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT_30x. 3. Auxiliary data with invalid Date of Birth data objects MUST have been sent by authorized terminal during Terminal Authentication mechanism. The date of birth MUST fit the required age, but an incorrect tag 0x72 (instead of 0x73) is used. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. Checking error in a valid Secure Messaging response.

3.15.14 Test case EAC2_ISO7816_U_14

Test – ID	EAC2_ISO7816_U_14

Purpose	Negative test with invalid Authentication Data Object containing an incorrect OID tag
Version	EAC2_1.1
Profile	eID, CMP
Preconditions	1. The PACE mechanism MUST have been performed using PIN, age verification and document verification must be allowed by CHAT 2. The Terminal Authentication mechanism MUST have been performed with the certificates defined in "Certificate Set 30" chapter as DV_CERT_30 and AT_CERT_30x. 3. Auxiliary data with invalid Date of Birth data objects MUST have been sent by authorized terminal during Terminal Authentication mechanism. The first date of birth MUST fit the required age, but an incorrect tag 0x07 (instead of 0x06) for OID is used. 4. The Chip Authentication MUST have been performed 5. The eID application MUST have been selected 6. All APDUs are sent as valid Secure Messaging APDUs
Test scenario	1. Send the given Compare APDU to the eID Card. '0C 33 00 00 <Lc> 85 <L_{85}> <Cryptogram> 8E 08 <Checksum> 00' <Cryptogram> contains the following encrypted data objects: <id-DateOfBirth>
Expected results	1. Checking error in a valid Secure Messaging response.

3.16 Unit test EAC2_ISO7816_V_Chip Authentication Version 3

This version of the Chip Authentication is an alternative to Chip Authentication Version 2 combined with Restricted Identification providing also an authentication of the sector-specific identifier towards the terminal and the pseudonymity of the eIDAS token without the need to use the same keys on several chips. Cryptographically, the protocol is based on the combination of an ephemeral key agreement with a Pseudonymous Signature (PS).

The EF.CardSecurity file may contain an optional key reference identifier. This is useful if the chip supports multiple keys for Chip Authentication. The MSE:Set AT command can be called either with implicit key selection if no key reference is included in EF.CardSecurity or with the explicit key reference defined in the EF.CardSecurity element. All tests in this unit SHOULD be used with implicit or explicit key reference depending on the presence of the key reference element in EF.CardSecurity.

The EF.CardSecurity may contain more than one ChipAuthenticationPublicKeyInfo. In this case, all appropriate tests MUST be performed for each key. The corresponding test case is only rated as a PASS if all test case runs are completed successfully. For test cases where the Chip Authentication mechanism is just used as precondition the first key is always used.

The tests defined in this test unit shall be run once. In Preconditions and otherwise specified, Terminal Authentication version 2 SHALL be performed with the certificates DV_CERT_17 and AT_CERT_17i.

The eIDAS token MAY return up to two sector-specific pseudonyms during execution of the Pseudonymous Signature Authentication (PSA). In the appropriate test cases of this Unit, the generation and output of each pseudonym shall conform to the PSAInfo information provided in the ICS. If not, the step of the scenario shall fail.

3.16.1 Test case EAC2_ISO7816_V_1

Test – ID	EAC2_ISO7816_V_1
Purpose	Positive test: Perform Chip Authentication Version 3 with MSE: Set AT / General Authenticate commands
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in the file EF.CardSecurity 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Restart Secure Messaging with new derived session keys

	4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' within Secure Messaging
Expected results	1. '90 00' within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. - 4. '90 00' within a valid Secure Messaging response 5. '7C <L_{7C}> 82 <L_{82}> <First Pseudonym Public Key> 83 <L_{83}> <Second Pseudonym Public Key> 84 <L_{84}> <Pseudonymous Signature> 90 00' in a valid Secure Messaging response. The presence of Tag 82 and Tag 83 shall be coherent with the ICS

3.16.2 Test case EAC2_ISO7816_V_2

Test – ID	EAC2_ISO7816_V_2
Purpose	Positive test: Perform Chip Authentication Version 3 with MSE: Set AT / General Authenticate commands The ephemeral PACE-GM Public key shall be reused by the chip. If not, the test is not applicable
Version	EAC2_1.1
Profile	eID, CA3_ReUse
Preconditions	1. ECDH Domain parameters between PACE-GM and CA3 must be identical 2. Perform PACE with Generic Mapping 3. Terminal Authentication Version 2 must be performed 4. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging

	3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with ephemeral public key as input. Use the ephemeral public key of PACE mapping (GM) as ephemeral public key. Perform Pseudonymous Signature Authentication with computed signature: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' within Secure Messaging
Expected results	1. '90 00' within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response. The Public Key shall be identical to the one sent during the step 2 of PACE GM mechanism 3. - 4. '90 00' within a valid Secure Messaging response 5. '7C <L_{7C}> 82 <L_{82}> <First Pseudonym Public Key> 83 <L_{83}> <Second Pseudonym Public Key> 84 <L_{84}> <Pseudonymous Signature> 90 00' in a valid Secure Messaging response. The presence of Tag 82 and Tag 83 shall be coherent with the ICS

3.16.3 Test case EAC2_ISO7816_V_3

Test – ID	EAC2_ISO7816_V_3
Purpose	Negative test: Perform Pseudonymous Signature command with a valid ephemeral public key, but without Secure Messaging
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging

	3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' without Secure Messaging
Expected results	1. '90 00' within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. - 4. '90 00' within a valid Secure Messaging response 5. Checking error or Execution error or Warning in plain response

3.16.4 Test case EAC2_ISO7816_V_4

Test – ID	EAC2_ISO7816_V_4
Purpose	Negative test: Perform Pseudonymous Signature command with a valid ephemeral public key, but with invalid data object tag for the sector public key.
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <Lc> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging

	5. Compute Pseudonymous Signature with ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature, using a wrong tag (84 instead of 80): '00 86 00 00 <L_c> 7C <L_{7C}> 84 <L_{84}> <sector public key> <L_e>' within Secure Messaging
Expected results	1. '90 00' within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. - 4. '90 00' within a valid Secure Messaging response 5. Checking error or Execution error or Warning in valid Secure Messaging response

3.16.5 Test case EAC2_ISO7816_V_5

Test – ID	EAC2_ISO7816_V_5
Purpose	Negative test: Perform Pseudonymous Signature command with a valid ephemeral public key, but with missing sector public key
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature but missing sector public key: '00 86 00 00 <L_c> 7C 00 <L_e>'

	within Secure Messaging
Expected results	1. ‘90 00’ within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. - 4. ‘90 00’ within a valid Secure Messaging response 5. Checking error or Execution error or Warning in valid Secure Messaging response

3.16.6 Test case EAC2_ISO7816_V_6

Test – ID	EAC2_ISO7816_V_6
Purpose	Negative test: Perform Pseudonymous Signature command with a valid ephemeral public key, but with an empty sector public key
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature but empty sector public key: '00 86 00 00 <L_c> 7C 03 80 01 00 <L_e>' within Secure Messaging
Expected results	1. ‘90 00’ within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. -

	4. '90 00' within a valid Secure Messaging response 5. Checking error or Execution error or Warning in valid Secure Messaging response

3.16.7 Test case EAC2_ISO7816_V_7

Test – ID	EAC2_ISO7816_V_7
Purpose	Negative test: Perform Chip Authentication Version 3 of the Anonymous Diffie-Hellman in the first part, but with old session keys during the Pseudonymous Signature Authentication
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Do not restart Secure Messaging and send the given MSE: Set AT APDU with old SM keys to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within secure messaging (old SM keys)
Expected results	1. '90 00' within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. Checking error or Execution error or Warning in plain

3.16.8 Test case EAC2_ISO7816_V_8

Test – ID	EAC2_ISO7816_V_8
Purpose	Negative test: Perform Chip Authentication Version 3 with a public key in the first General Authenticate command that does not match the compressed public key sent during Terminal Authentication
Version	EAC2_1.1
Profile	eID, CA3

Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within secure messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging.Use an ephemeral public key that does not match to the compressed public key sent in Terminal Authentication
Expected results	1. ‘90 00’ within a valid Secure Messaging response 2. Checking error or Execution error or Warning in valid Secure Messaging response

3.16.9 Test case EAC2_ISO7816_V_9

Test – ID	EAC2_ISO7816_V_9
Purpose	Negative test: Perform Chip Authentication Version 3 with a public key in the first General Authenticate command that is not on the curve
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within secure messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging. Use an ephemeral public key that does not belong to the curve.
Expected results	1. ‘90 00’ within a valid Secure Messaging response 2. Checking error or Execution error or Warning in valid Secure Messaging response

3.16.10 Test case EAC2_ISO7816_V_10

Test – ID	EAC2_ISO7816_V_10
Purpose	Negative test: Perform Chip Authentication Version 3 without Anonymous Diffie-Hellman key agreement before Pseudonymous Signature Authentication.
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within secure messaging. 2. Compute Pseudonymous Signature. Perform Pseudonymous Signature Authentication with computed signature: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' within secure messaging
Expected results	1. ‘90 00’ within a valid Secure Messaging response 2. If CA2 is NOT supported: Checking error or Execution error or Warning in valid Secure Messaging response If CA2 is supported: '90 00' in a valid Secure Messaging response

3.16.11 Test case EAC2_ISO7816_V_11

Test – ID	EAC2_ISO7816_V_11
Purpose	Positive test: Perform Chip Authentication Version 3 commands in an unusual sequence: commands of ADH are performed once and commands of PSA are performed a second time
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity 2. Send the given General Authenticate APDU to the eIDAS token:

	'00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with a correct ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' within Secure Messaging 6. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 7. Compute Pseudonymous Signature with a correct ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' within Secure Messaging
Expected results	1. '90 00' within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. - 4. '90 00' within a valid Secure Messaging response 5. '7C <L_{7C}> 82 <L_{82}> <First Pseudonym Public Key> 83 <L_{83}> <Second Pseudonym Public Key> 84 <L_{84}> <Pseudonymous Signature> 90 00' in a valid Secure Messaging response. The presence of Tag 82 and Tag 83 shall be coherent with the ICS 6. '90 00' within a valid Secure Messaging response 7. '7C <L_{7C}> 82 <L_{82}> <First Pseudonym Public Key> 83 <L_{83}> <Second Pseudonym Public Key> 84 <L_{84}> <Pseudonymous Signature> 90 00' in a valid Secure Messaging response. The presence of Tag 82 and Tag 83 shall be coherent with the ICS

3.16.12 Test case EAC2_ISO7816_V_12

Test – ID	EAC2_ISO7816_V_12

Purpose	Negative test: Perform Chip Authentication Version 3 commands where PSA is using a sector-specific key that does not match to the hash of the certificate extension
Version	EAC2_1.1
Profile	eID, CA3
Preconditions	1. Perform PACE with Generic Mapping 2. Terminal Authentication Version 2 must be performed 3. Extract ChipAuthenticationPublicKeyInfo from EF.CardSecurity and generate an ephemeral key pair
Test scenario	1. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <CA OID> 84 <L_{84}> <private key reference>' within Secure Messaging. The private key reference MUST be included in the APDU specified in the ChipAuthenticationInfo structure stored in EF.CardSecurity file 2. Send the given General Authenticate APDU to the eIDAS token: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <ephemeral public key> <L_e>' within Secure Messaging 3. Restart Secure Messaging with new derived session keys 4. Send the given MSE: Set AT APDU to the eIDAS token: '00 22 41 A4 <L_c> 80 <L_{80}> <PSA OID> 84 <L_{84}> <private key reference>' within Secure Messaging 5. Compute Pseudonymous Signature with a correct ephemeral public key as input. Perform Pseudonymous Signature Authentication with computed signature. Use a sector-specific key that does not match to hash of certificate extension: '00 86 00 00 <L_c> 7C <L_{7C}> 80 <L_{80}> <sector public key> <L_e>' within Secure Messaging
Expected results	1. ‘90 00’ within a valid Secure Messaging response 2. '7C <L_{7C}> 81 <L_{81}> <Public Key> 90 00' in a valid Secure Messaging response 3. - 4. ‘90 00’ within a valid Secure Messaging response 5. 'Checking error or Execution error or Warning in valid Secure Messaging response

4 Tests for layer 7 (Data Structure)

4.1 Unit EAC2_DATA_A, EF.CardAccess

This unit covers all tests about the coding of the elementary file EF.CardAccess containing relevant data for establishing the security protocols PACE, CA and TA.

4.1.1 Test case EAC2_DATA_A_1a

Test - ID	EAC2_DATA_A_1a
Purpose	Test the ASN.1 encoding of the SecurityInfos (PACE)
Version	EAC2_1.03
Profile	PACE, TA2, (CA2 or CA3)
Preconditions	1. EF.CardAccess MUST have been read from the eID Card
Test scenario	1. The content of the SecurityInfos object MUST be encoded according to the SecurityInfos syntax definition. 2. At least one PACEInfo object MUST exist 3. For each supported set of proprietary PACE domain parameters a PACEDomainParameterInfo object MUST exist 4. At least one TerminalAuthenticationInfo MUST exist 5. Exactly one CardInfo MUST exist
Expected results	1. true 2. true 3. true 4. true 5. true

4.1.2 Test case EAC2_DATA_A_1b

Test - ID	EAC2_DATA_A_1b
Purpose	Test the ASN.1 encoding of the SecurityInfos (CA)
Version	EAC2_1.03
Profile	PACE, TA2, (CA2 or CA3), CSTA
Preconditions	1. EF.CardAccess MUST have been read from the eID Card
Test scenario	1. The content of the SecurityInfos object MUST be encoded according to the SecurityInfos syntax definition. 2. At least one ChipAuthenticationInfo object MUST exist 3. At least one ChipAuthenticationDomainParameterInfo MUST exist
Expected results	1. true

	2. true 3. true

4.1.3 Test case EAC2_DATA_A_2

Test - ID	EAC2_DATA_A_2
Purpose	Test the ASN.1 encoding of the PACEInfo
Version	EAC2_1.03
Profile	PACE
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed and this test is repeated for each PACEInfo element containing the OID specified in the EAC 2.0 specification [R8] and the version element set to 2.
Test scenario	1. The PACEInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. If standardized domain parameters are used the parameterID MUST reference a valid standardized domain parameter. If multiple proprietary domain parameters are used the parameterId reference in the PACEInfo MUST be coherent with the ICS (See A) and there MUST be a corresponding PACEDomainParameterInfo with compatible protocol OID (e.g. both contain DH-GM)
Expected results	1. true 2. true

4.1.4 Test case EAC2_DATA_A_3

Test - ID	EAC2_DATA_A_3
Purpose	Test the ASN.1 encoding of the PACEDomainParameterInfo This test case MUST be performed if proprietary domain parameters are used. If standardized domain parameters are used this test case MUST NOT be performed.
Version	EAC2_1.03
Profile	PACE
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed and this test is repeated for each PACEDomainParameterInfo element containing the OID specified in the EAC 2.0 specification [R8].
Test scenario	1. The PACEDomainParameterInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. The presence of the parameterId reference in the PACEDomainParameterInfo MUST be coherent with the ICS (See A) and there MUST be a corresponding PACEInfo with compatible protocol OID (e.g. both contain DH-GM) 3. If proprietary domain parameters are used the algorithm identifier domainParameter MUST be suitable to the key agreement protocol and its

	algorithm OID MUST be one of the following: • dhpublicnumber (OID: 1.2.840.10046.2.1) • id-ecPublicKey (OID: 1.2.840.10045.2.1) 4. The algorithm identifier's parameters MUST follow X9.42 (DH) [R11] or ECC specification (ECDH) [R6] and MUST be valid.
Expected results	1. true 2. true 3. true 4. true

4.1.5 Test case EAC2_DATA_A_4

Test - ID	EAC2_DATA_A_4
Purpose	Test the ASN.1 encoding of the ChipAuthenticationInfo
Version	EAC2_1.0
Profile	(CA2 or CA3)
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed and this test is repeated for each ChipAuthenticationInfo element containing the OID specified in the EAC specification [R8] and the version element set to 2. If there is no ChipAuthenticationInfo element available in EF.CardAccess, this test case can be skipped.
Test scenario	1. The ChipAuthenticationInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. The presence of the keyId reference in the ChipAuthenticationInfo MUST be coherent with the ICS (See A)
Expected results	1. true 2. true

4.1.6 Test case EAC2_DATA_A_5

Test - ID	EAC2_DATA_A_5
Purpose	Test the ASN.1 encoding of the ChipAuthenticationDomainParameterInfo
Version	EAC2_1.03
Profile	(CA2 or CA3)
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed and this test is repeated for each ChipAuthenticationDomainParameterInfo element containing the OID specified in the EAC 2.0 specification [R8]. If there is no ChipAuthenticationDomainParameterInfo element available in EF.CardAccess, this test case can be skipped.
Test scenario	1. The ChipAuthenticationDomainParameterInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. The presence of the keyId reference in the

	ChipAuthenticationDomainParameterInfo MUST be coherent with the ICS (See A) and there MUST be a corresponding ChipAuthenticationInfo with compatible protocol OID (e.g. both contain DH) 3. The algorithm identifier domainParamter MUST contain as parameters a valid Integer as specified in [R8] if standardized domain parameters are used. If proprietary domain parameters are used the algorithm identifier domainParameter MUST be suitable to the key agreement protocol and its algorithm OID MUST be one of the following: • dhpublicnumber (OID: 1.2.840.10046.2.1) • id-ecPublicKey (OID: 1.2.840.10045.2.1) 4. The algorithm identifier's parameters MUST follow X9.42 (DH) [R11] or ECC specification (ECDH) [R6] and MUST be valid.
Expected results	1. true 2. true 3. true 4. true

4.1.7 Test case EAC2_DATA_A_6

Test - ID	EAC2_DATA_A_6
Purpose	Test the ASN.1 encoding of the TerminalAuthenticationInfo
Version	EAC2_1.0
Profile	TA2
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed and this test is repeated for each TerminalAuthenticationInfo element containing the OID specified in the EAC 2.0 specification [R8] and the version element set to 2.
Test scenario	1. The TerminalAuthenticationInfo element must follow the ASN.1 syntax definition in the EAC specification [R8].
Expected results	1. true

4.1.8 Test case EAC2_DATA_A_7

Test - ID	EAC2_DATA_A_7
Purpose	Test the ASN.1 encoding of the CardInfo
Version	EAC2_1.0
Profile	CardInfo
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed
Test scenario	1. The CardInfo element must follow the ASN.1 syntax definition in the EAC specification [R8].
Expected results	1. true

4.1.9 Test case EAC2_DATA_A_8

Test - ID	EAC2_DATA_A_8
Purpose	Test the ASN.1 encoding of the PSAInfo
Version	EAC2_1.0
Profile	PSAInfo
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed
Test scenario	1. The PSAInfo element must follow the ASN.1 syntax definition in the EAC specification [R8].
Expected results	1. true

4.1.10 Test case EAC2_DATA_A_9

Test - ID	EAC2_DATA_A_9
Purpose	Test the ASN.1 encoding of the ProvilegedTerminalInfo
Version	EAC2_1.0
Profile	PrivTerInfo
Preconditions	1. Test case EAC2_DATA_A_1 MUST have been performed 2. The data object containing SecurityInfos is parsed
Test scenario	1. The PrivilegedTerminalInfo element must follow the ASN.1 syntax definition in the EAC specification [R8].
Expected results	1. true

4.2 Unit EAC2_DATA_B, EF.CardSecurity

This unit covers all tests about the coding of the elementary file EF.CardSecurity containing the full set of data for establishing the security protocols PACE, CA and TA. This file is digitally signed.

4.2.1 Test case EAC2_DATA_B_1

Test - ID	EAC2_DATA_B_1
Purpose	Test the ASN.1 encoding of the SecurityInfos in EF.CardSecurity
Version	EAC2_1.03
Profile	PACE, TA2, (CA2 or CA3)
Preconditions	1. EF.CardSecurity MUST have been read from the eID Card
Test scenario	1. The content of the SecurityInfos object MUST be encoded according to the SecurityInfos syntax definition. 2. EF.CardSecurity MUST be implemented as SignedData according to the EAC specification [R8]. 3. The signature MUST be verified. 4. At least one PACEInfo object MUST exist

	5. For each supported set of proprietary PACE domain parameters a PACEDomainParameterInfo object MUST exist 6. At least one ChipAuthenticationInfo object MUST exist 7. At least one ChipAuthenticationDomainParameterInfo MUST exist 8. At least one ChipAuthenticationPublicKeyInfo MUST exist 9. At least one TerminalAuthenticationInfo MUST exist 10. Exactly one CardInfoLocator MUST exist
Expected results	1. true 2. true 3. true 4. true 5. true 6. true 7. true 8. true 9. true 10. true

4.2.2 Test cases EAC2_DATA_B_2 to EAC2_DATA_B_7

Test cases EAC2_DATA_B_2 to EAC2_DATA_B_7 are equally performed on SecurityInfo objects from EF.CardSecurity like test cases EAC2_DATA_A_2 to EAC2_DATA_A_7 were performed on SecurityInfo objects EF.CardAccess before. References to EAC2_DATA_A_1 are replaced by references to EAC2_DATA_B_1. The profile CSTA is only relevant for test suite EAC2_DATA_A but not for test suite EAC2_DATA_B. Also the conditions of test cases EAC2_DATA_A_4 and EAC2_DATA_A_5 are only relevant for test suite EAC2_DATA_A but not for test suite EAC2_DATA_B.

4.2.3 Test case EAC2_DATA_B_8

Test - ID	EAC2_DATA_B_8
Purpose	Test the ASN.1 encoding of the ChipAuthenticationPublicKeyInfo
Version	EAC2_1.03
Profile	(CA2 or CA3)
Preconditions	1. Test case EAC2_DATA_B_1 MUST have been performed 2. The data object containing SecurityInfos is parsed and this test is repeated for each ChipAuthenticationPublicKeyInfo element containing the OID specified in the EAC 2.0 specification [R8].
Test scenario	1. The ChipAuthenticationPublicKeyInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. The presence of the keyId reference in the ChipAuthenticationPublicKeyInfo MUST be coherent with the ICS (See Annex A) and there MUST be corresponding ChipAuthenticationInfo and ChipAuthenticationDomainParameterInfo with compatible protocol OID (e.g. all contain DH)

	3. The algorithm identifier MUST contain as parameters a valid Integer as specified in [R8] if standardized domain parameters are used. If proprietary domain parameters are used the algorithm identifier MUST be suitable to the key agreement protocol and its algorithm OID MUST be one of the following: • dhpublicnumber (OID: 1.2.840.10046.2.1) • id-ecPublicKey (OID: 1.2.840.10045.2.1) 4. The algorithm identifier's parameters MUST follow X9.42 (DH) [R11] or ECC specification (ECDH) [R6] and MUST be valid.
Expected results	1. true 2. true 3. true 4. true

4.2.4 Test case EAC2_DATA_B_9

Test - ID	EAC2_DATA_B_9
Purpose	Test the ASN.1 encoding of the RestrictedIdentificationInfo
Version	EAC2_1.0
Profile	RI
Preconditions	1. Test case EAC2_DATA_B_1 MUST have been performed and at least one RestrictedIdentificationInfo object MUST exist 2. The data object containing SecurityInfos is parsed and this test is repeated for each RestrictedIdentificationInfo element containing the OID specified in the EAC specification [R8] and the version element set to 1.
Test scenario	1. The RestrictedIdentificationInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. The presence of the keyId reference in the RestrictedIdentificationInfo MUST be coherent with the ICS (See Annex A)
Expected results	1. true 2. true

4.2.5 Test case EAC2_DATA_B_10

Test - ID	EAC2_DATA_B_10
Purpose	Test the ASN.1 encoding of the RestrictedIdentificationDomainParameterInfo
Version	EAC2_1.03
Profile	RI_DP
Preconditions	1. Test case EAC2_DATA_B_1 MUST have been performed and exactly one RestrictedIdentificationDomainParameterInfo object MUST exist 2. The data object containing SecurityInfos is parsed.
Test scenario	1. The RestrictedIdentificationDomainParameterInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. The algorithm identifier domainParamter MUST contain as parameters a

	valid Integer as specified in [R8] if standardized domain parameters are used. If proprietary domain parameters are used the algorithm identifier domainParameter MUST be suitable to the key agreement protocol and its algorithm OID MUST be one of the following: • dhpublicnumber (OID: 1.2.840.10046.2.1) • id-ecPublicKey (OID: 1.2.840.10045.2.1) 3. The algorithm identifier's parameters MUST follow X9.42 (DH) [R11] or ECC specification (ECDH) [R6] and MUST be valid.
Expected results	1. true 2. true 3. true

4.2.6 Test case EAC2_DATA_B_11

Test - ID	EAC2_DATA_B_11
Purpose	Test the coherency between EF.CardSecurity and EF.CardAccess
Version	EAC2_1.1
Profile	
Preconditions	1. EF.CardAccess MUST have been read successfully 2. EF.CardSecurity MUST have been read successfully
Test scenario	1. Check the SecurityInfo structures stored in EF.CardAccess are duplicated in the EF.CardSecurity
Expected results	1. Each SecurityInfo structure stored in the EF.CardAccess is also present in EF.CardSecurity (EF.CardAccess is a subset of EF.CardSecurity)

4.3 Unit EAC2_EIDDATA_B eID Data Groups

This unit covers all tests about the coding of the elementary files of the eID application. Due to the simplicity of the encoded elements all data groups are tested within one test unit. Not all data groups must be present in all cases of implementation, therefore only the tests fitting the eID Card personalization must be performed.

4.3.1 Test case EAC2_EIDDATA_B_1

Test - ID	EAC2_EIDDATA_B_1
Purpose	Test the ASN.1 encoding of the eID DG1 elementary file
Version	EAC2_1.0
Profile	eID, DG1
Preconditions	1. DG1 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the DocumentType syntax definition.
Expected results	1. true

4.3.2 Test case EAC2_EIDDATA_B_2

Test - ID	EAC2_EIDDATA_B_2
Purpose	Test the ASN.1 encoding of the eID DG2 elementary file
Version	EAC2_1.0
Profile	eID, DG2
Preconditions	1. DG2 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the IssuingState syntax definition.
Expected results	1. true

4.3.3 Test case EAC2_EIDDATA_B_3

Test - ID	EAC2_EIDDATA_B_3
Purpose	Test the ASN.1 encoding of the eID DG3 elementary file
Version	EAC2_1.0
Profile	eID, DG3
Preconditions	1. DG3 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the DateOfExpiry syntax definition.
Expected results	1. true

4.3.4 Test case EAC2_EIDDATA_B_4

Test - ID	EAC2_EIDDATA_B_4
Purpose	Test the ASN.1 encoding of the eID DG4 elementary file
Version	EAC2_1.0
Profile	eID, DG4
Preconditions	1. DG4 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the GivenNames syntax definition.
Expected results	1. true

4.3.5 Test case EAC2_EIDDATA_B_5

Test - ID	EAC2_EIDDATA_B_5
Purpose	Test the ASN.1 encoding of the eID DG5 elementary file
Version	EAC2_1.0
Profile	eID, DG5
Preconditions	1. DG5 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the FamilyNames syntax definition.
Expected results	1. true

4.3.6 Test case EAC2_EIDDATA_B_6

Test - ID	EAC2_EIDDATA_B_6
Purpose	Test the ASN.1 encoding of the eID DG6 elementary file
Version	EAC2_1.0
Profile	eID, DG6
Preconditions	1. DG6 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the ArtisticName syntax definition.
Expected results	1. true

4.3.7 Test case EAC2_EIDDATA_B_7

Test - ID	EAC2_EIDDATA_B_7
Purpose	Test the ASN.1 encoding of the eID DG7 elementary file
Version	EAC2_1.0
Profile	eID, DG7
Preconditions	1. DG7 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the AcademicTitle syntax definition.
Expected results	1. true

4.3.8 Test case EAC2_EIDDATA_B_8

Test - ID	EAC2_EIDDATA_B_8
Purpose	Test the ASN.1 encoding of the eID DG8 elementary file
Version	EAC2_1.0
Profile	eID, DG8
Preconditions	1. DG8 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the DateOfBirth syntax definition.
Expected results	1. true

4.3.9 Test case EAC2_EIDDATA_B_9

Test - ID	EAC2_EIDDATA_B_9
Purpose	Test the ASN.1 encoding of the eID DG9 elementary file
Version	EAC2_1.0
Profile	eID, DG9
Preconditions	1. DG9 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the PlaceOfBirth syntax definition.
Expected results	1. true

4.3.10 Test case EAC2_EIDDATA_B_10

Test - ID	EAC2_EIDDATA_B_10
Purpose	Test the ASN.1 encoding of the eID DG10 elementary file
Version	EAC2_1.0
Profile	eID, DG10
Preconditions	1. DG10 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the Nationality syntax definition.
Expected results	1. true

4.3.11 Test case EAC2_EIDDATA_B_11

Test - ID	EAC2_EIDDATA_B_11
Purpose	Test the ASN.1 encoding of the eID DG11 elementary file
Version	EAC2_1.0
Profile	eID, DG11
Preconditions	1. DG11 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the Sex syntax definition.
Expected results	1. true

4.3.12 Test case EAC2_EIDDATA_B_12

Test - ID	EAC2_EIDDATA_B_12
Purpose	Test the ASN.1 encoding of the eID DG12 elementary file
Version	EAC2_1.0
Profile	eID, DG12
Preconditions	1. DG12 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the OptionalDataR syntax definition.
Expected results	1. true

4.3.13 Test case EAC2_EIDDATA_B_13

Test - ID	EAC2_EIDDATA_B_13
Purpose	Test the ASN.1 encoding of the eID DG17 elementary file
Version	EAC2_1.0
Profile	eID, DG17
Preconditions	1. DG17 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the PlaceOfResidence syntax definition.
Expected results	1. true

4.3.14 Test case EAC2_EIDDATA_B_14

Test - ID	EAC2_EIDDATA_B_14
Purpose	Test the ASN.1 encoding of the eID DG18 elementary file
Version	EAC2_1.0
Profile	eID, DG18
Preconditions	1. DG18 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the MunicipalityID syntax definition.
Expected results	1. true

4.3.15 Test case EAC2_EIDDATA_B_15

Test - ID	EAC2_EIDDATA_B_15
Purpose	Test the ASN.1 encoding of the eID DG19 elementary file
Version	EAC2_1.0
Profile	eID, DG19
Preconditions	1. DG19 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the ResidencePermitI syntax definition.
Expected results	1. true

4.3.16 Test case EAC2_EIDDATA_B_16

Test - ID	EAC2_EIDDATA_B_16
Purpose	Test the ASN.1 encoding of the eID DG20 elementary file
Version	EAC2_1.0
Profile	eID, DG20
Preconditions	1. DG20 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the ResidencePermitII syntax definition.
Expected results	1. true

4.3.17 Test case EAC2_EIDDATA_B_17

Test - ID	EAC2_EIDDATA_B_17
Purpose	Test the ASN.1 encoding of the eID DG21 elementary file
Version	EAC2_1.0
Profile	eID, DG21
Preconditions	1. DG21 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the PhoneNumber syntax definition.
Expected results	1. true

4.3.18 Test case EAC2_EIDDATA_B_18

Test - ID	EAC2_EIDDATA_B_18
Purpose	Test the ASN.1 encoding of the eID DG22 elementary file
Version	EAC2_1.1
Profile	eID, DG22
Preconditions	1. DG22 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the EMailAddress syntax definition.
Expected results	1. true

4.3.19 Test case EAC2_EIDDATA_B_19

Test - ID	EAC2_EIDDATA_B_19
Purpose	Test the ASN.1 encoding of the eID DG13 elementary file
Version	EAC2_1.1
Profile	eID, DG13
Preconditions	1. DG13 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the BirthName syntax definition.
Expected results	1. true

4.3.20 Test case EAC2_EIDDATA_B_20

Test - ID	EAC2_EIDDATA_B_20
Purpose	Test the ASN.1 encoding of the eID DG14 elementary file
Version	EAC2_1.1
Profile	eID, DG14
Preconditions	1. DG14 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the WrittenSignature syntax definition.
Expected results	1. true

4.3.21 Test case EAC2_EIDDATA_B_21

Test - ID	EAC2_EIDDATA_B_21
Purpose	Test the ASN.1 encoding of the eID DG15 elementary file
Version	EAC2_1.1
Profile	eID, DG15
Preconditions	1. DG15 MUST have been read from the eID Card
Test scenario	1. The content of the data object MUST be encoded according to the Date syntax definition.
Expected results	1. true

4.4 Unit EAC2_DATA_C, EF.ChipSecurity

This unit covers all tests about the coding of the elementary file EF.ChipSecurtiy containing the signed SecurityInfos supported by the MRTD chip. This file SHALL be restricted to privileged terminals.

4.4.1 Test case EAC2_DATA_C_1

Test - ID	EAC2_DATA_C_1
Purpose	Test the ASN.1 encoding of the SecurityInfos in EF.ChipSecurity
Version	EAC2_1.1
Profile	CS
Preconditions	1. EF.ChipSecurity MUST have been read from the eID Card
Test scenario	1. The content of the SecurityInfos object MUST be encoded according to the SecurityInfos syntax definition. 2. EF. ChipSecurity MUST be implemented as SignedData according to the EAC specification [R8]. 3. The signature MUST be verified. 4. At least one PACEInfo object MUST exist 5. For each supported set of proprietary PACE domain parameters a PACEDomainParameterInfo object MUST exist 6. At least one ChipAuthenticationInfo object MUST exist 7. At least one ChipAuthenticationDomainParameterInfo MUST exist 8. At least one ChipAuthenticationPublicKeyInfo MUST exist 9. At least one TerminalAuthenticationInfo MUST exist 10. Exactly one CardInfoLocator MUST exist 11. Exactly one PrivilegedTerminalInfo MUST exist
Expected results	1. true 2. true 3. true 4. true 5. true 6. true 7. true 8. true 9. true 10. true 11. true

4.4.2 Test cases EAC2_DATA_C_2 to EAC2_DATA_C_10

Test cases EAC2_DATA_C_2 to EAC2_DATA_C_7 are equally performed on SecurityInfo objects from

EF.ChipSecurity like test cases EAC2_DATA_A_2 to EAC2_DATA_A_7 were performed on SecurityInfo objects in EF.CardAccess before. References to EAC2_DATA_A_1 are replaced by references to EAC2_DATA_C_1.
Test case EAC2_DATA_C_8 to EAC2_DATA C10 are equally performed on SecurityInfo objects from EF.ChipSecurity like test cases EAC2_DATA_B_8 toEAC2_DATA_B10 were performed on SecurityInfo objects in EF.CardSecurity before. References to EAC2_DATA_B_1 are replaced by references to EAC2_DATA_C_1.
The profile CSTA is only relevant for test suite EAC2_DATA_A but not for test suite EAC2_DATA_C. Also the conditions of test cases EAC2_DATA_A_4 and EAC2_DATA_A_5 are only relevant for test suite EAC2_DATA_A but not for test suite EAC2_DATA_C.

4.4.3 Test case EAC2_DATA_C_11

Test - ID	EAC2_DATA_C_11
Purpose	Test the ASN.1 encoding of the PrivilegedTerminalInfo
Version	EAC2_1.1
Profile	CS
Preconditions	1. Test case EAC2_DATA_C_1 MUST have been performed and exactly one PrivilegedTerminalInfo object MUST exist 2. The data object containing SecurityInfos is parsed.
Test scenario	1. The eIDSecurityInfo element must follow the ASN.1 syntax definition in the EAC specification [R8]. 2. For each ChipAuthenticationInfo encapsulated in PrivilegedTerminalInfo, the corresponding ChipAuthenticationPublicKeyInfo MUST also be included in PrivilegedTerminalInfo. 3. The presence of the keyId reference in the PrivilegedTerminalInfo MUST be coherent with the ICS (See Annex A)
Expected results	1. true 2. true 3. true

4.4.4 Test case EAC2_DATA_C_12

Test - ID	EAC2_DATA_C_11
Purpose	Test the ASN.1 encoding of the eIDSecurityInfo
Version	EAC2_1.1
Profile	CS
Preconditions	1. Test case EAC2_DATA_C_1 MUST have been performed and exactly one eIDSecurityInfo object MUST exist 2. The data object containing SecurityInfos is parsed.
Test scenario	1. The eIDSecurityInfo element must follow the ASN.1 syntax definition in the EAC specification [R8].
Expected results	1. true

Annex A Implementation conformance statement

In order to set up the tests properly, an applicant SHALL provide the information specified in this annex. Some tests defined in this document are depending on the supported functionality of the eID Card. The test results will only cover the function declared in this statement.

A.1 Supported profiles

Tests that require functions not supported by the provided eID Card will be skipped during the tests. Please specify the profiles supported by the provided sample. For details on the profiles, please refer to section 2.2.

Application Profile	Applicant declaration (YES or NO)
ePassport	
eID	
eSign	

Protocol Profile	Applicant declaration (YES or NO)
Migration of the cryptographic system	
Certificate date validation	
Restricted Identification Domain Parameters	
Auxiliary Data Verification	
Change PIN after PACE using PUK allowed	
Change PIN for authentication terminals with “PIN Management” access rights allowed	
Change CAN for authentication terminals with “PIN Management” access rights allowed	
BAC	
PACE	
TA1	
TA2	
CA1	
CA2	
CA3	
CA3_ReUse	
Chip Security	
Envelope mechanism (ENV)	
Compare (CMP)	
AUTH_EXT	
CSTA	
PSAInfo	
CardInfo	
PrivTerInfo	

Algorithm Profile	Applicant declaration (YES or NO)
For Terminal Authentication based on ECDSA algorithm, include domain parameter in link certificate (LINK_CERT_7, LINK_CERT_8, LINK_CERT_9)	

DG	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
ePassport																						
eID																						

A.2 Supported cryptographic algorithm

The applicant of the passport under test SHALL declare the cryptosystem (signature algorithm and hash algorithm) used to perform the Terminal Authentication.

Signature algorithm	Key size (incl. curve name for ECDSA)	Hash algorithm

A.3 Cryptosystem migration policy

If the eID Card under test supports the migration to another cryptosystem, the applicant SHALL provide the list of supported target(s) cryptosystem(s) (signature algorithm and hash algorithm).

Note: For each target algorithm specified in this table, the test unit EAC2_ISO7816_N has to be performed. Afterward, the complete test set has to be repeated for each new algorithm.

Signature algorithm	Key size (incl. curve name for ECDSA)	Hash algorithm

A.4 EF.CardSecurity information

The applicant SHALL declare all supported protocol suites. The EF.CardSecurity file SHALL contain all necessary SecurityInfo objects. Algorithm profiles like DH/ECDH are directly derived from this table.

Protocol Suite	Applicant declaration (YES or NO)

id-PACE-DH-GM-3DES-CBC-CBC	
id-PACE-DH-GM-AES-CBC-CMAC-128	
id-PACE-DH-GM-AES-CBC-CMAC-192	
id-PACE-DH-GM-AES-CBC-CMAC-256	
id-PACE-ECDH-GM-3DES-CBC-CBC	
id-PACE-ECDH-GM-AES-CBC-CMAC-128	
id-PACE-ECDH-GM-AES-CBC-CMAC-192	
id-PACE-ECDH-GM-AES-CBC-CMAC-256	
id-PACE-DH-IM-3DES-CBC-CBC	
id-PACE-DH-IM-AES-CBC-CMAC-128	
id-PACE-DH-IM-AES-CBC-CMAC-192	
id-PACE-DH-IM-AES-CBC-CMAC-256	
id-PACE-ECDH-IM-3DES-CBC-CBC	
id-PACE-ECDH-IM-AES-CBC-CMAC-128	
id-PACE-ECDH-IM-AES-CBC-CMAC-192	
id-PACE-ECDH-IM-AES-CBC-CMAC-256	
id-PACE-ECDH-CAM-AES-CBC-CMAC-128	
id-PACE-ECDH-CAM-AES-CBC-CMAC-192	
id-PACE-ECDH-CAM-AES-CBC-CMAC-256	
id-CA-DH-3DES-CBC-CBC	
id-CA-DH-AES-CBC-CMAC-128	
id-CA-DH-AES-CBC-CMAC-192	
id-CA-DH-AES-CBC-CMAC-256	
id-CA-ECDH-3DES-CBC-CBC	
id-CA-ECDH-AES-CBC-CMAC-128	
id-CA-ECDH-AES-CBC-CMAC-192	
id-CA-ECDH-AES-CBC-CMAC-256	
id-RI-DH-SHA-1	
id-RI-DH-SHA-224	
id-RI-DH-SHA-256	
id-RI-ECDH-SHA-1	
id-RI-ECDH-SHA-224	
id-RI-ECDH-SHA-256	

If the eID Card under test supports Restricted Identification, the applicant SHALL provide all available private keys. There SHALL be at least one key with “authorized only” attribute set to YES and vice versa.

Note: For each key specified in this table, the test unit EAC2_ISO7816_R has to be performed.

Restricted Identification (public key)	Key ID	authorized only (YES or NO)

A.5 Additional Information

PIN	
Minimum PIN length	
PUK	
Default Retry Counter	
Valid Municipality ID	
Valid Age	
Invalid password reference for MSE:SetAT command at the beginning of PACE protocol (see Test case EAC2_ISO7816_H_23)	
Invalid private key reference for MSE:SetAT command at the beginning of CA protocol (see Test case EAC2_ISO7816_I_14)	
Command to send to the eID card to verify the chip's ability to still require Secured APDU. If not provided, use '00 B0 81 00 00'.	

A.6 PSA Information

The applicant has to provide following information about ps1-authInfo and ps2-authInfo in the PSAInfo supported by the eID card.

Supported ps1-authInfo and ps2-authInfo:

Data structure	Protocol	KeyId	ps1-authInfo value	ps2-authInfo value

PSA cipher suites:

Protocol Suite	Applicant declaration (YES or NO)
id-PSA-ECDH-ECSchnorr-SHA-256	
id-PSA-ECDH-ECSchnorr-SHA-384	
id-PSA-ECDH-ECSchnorr-SHA-512	