3GPP TS 31.048 V5.1.0 (2005-10)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Terminals;
Security mechanisms for the (U)SIM application toolkit;
Test specification
(Release 5)

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Keywords
GSM, UMTS, SIM, API

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document covers the minimum characteristics considered as necessary in order to provide compliance to 3GPP TS 23.048 " Security Mechanisms for the (U)SIM application toolkit; Stage 2" [6].

The present document describes the technical characteristics and methods of test for testing the Security Mechanisms for the (U)SIM application toolkit. It specifies the following parts:

- test applicability
- test environment description
- tests format
- test area reference
- conformance requirements
- Test Area Files
- test procedure
- test coverage
- a description of the associated testing tools that shall be used.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.040: "Technical realization of the Short Message Service (SMS)".

[3] 3GPP TS 51.011 Release 4: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface".

[4] 3GPP TS 31.111: "USIM Application Toolkit (USAT)".

[5] 3GPP TS 43.019: "Subscriber Identity Module Application Programming Interface (SIM API); SIM API for Java Card™; Stage 2".

[6] 3GPP TS 23.048: "Digital cellular telecommunications system (Phase 2+); Security Mechanisms for the SIM application toolkit; Stage 2"

[7] SUN Java Card Specification "Java Card 2.1 VM Architecture Specification".

[8] ETSI TS 101 220 Release 5: "Numbering System for Telecommunication IC card applications".

[9] 3GPP TS 51.013: "Test specification for Subscriber Identity Module (SIM) Application Programming Interface (API) for Java Card™".

[10] 3GPP TS 23.041: "Technical realization of Cell Broadcast Service (CBS)".

3 Definitions and abbreviations

3.1 Definitions

Applet: An Applet is an application built up using a number of classes which will run under the control of the Java Card virtual machine.

Applet installation parameters: Default values for applet installation parameters.

Applet loading script: File containing the APDU commands that will load and install the test applet in the card.

CleanUp Script file: File containing the APDU commands that will restore the Default Initial Conditions on the SIM

Conformance Requirement Reference: Description of the expected card behaviour according to TS 23.048 [6].

Expected state: the state in which the (U)SIM is supposed to be after the execution of the test procedure applied on the relevant initial conditions

Security parameters: Minimum security requirements defined for the applet installation process.

Test Area: Set of Test Cases applicable to a specific part (Security mechanisms, Remote file management, …) of the TS 23.048 [6].

Test Case: Elementary test that checks the compliance with one or more Conformance Requirement References.

Test procedure: the sequence of actions/commands to perform all the test cases defined in a test area.

Test Script file: File containing the APDU commands that will execute and verify the test results.

Test Applet: Applet designed to test a specific functionality of the TS 23.048 [6].

3.2 Abbreviations

For the purpose of the present document, the following abbreviations apply, in addition to those listed in TR 21.905 [1]:

AC	Application Code
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
CAD	Card Acceptance Device
FFS	For Further Study
IFD	Interface Device
JCRE	Java Card™ Run Time Environment
JVM	Java Virtual Machine
SIM	Subscriber Identity Module
SE	Sending Entity
SPI1	First byte of SPI field
SPI2	Second byte SPI field
SD	Secured Data
ARD	Additional Response Data
SC	Status Code

4 Test Environment

This clause specifies requirements that shall be met and the testing rules that shall be followed during the test procedure.

4.1 Applicability

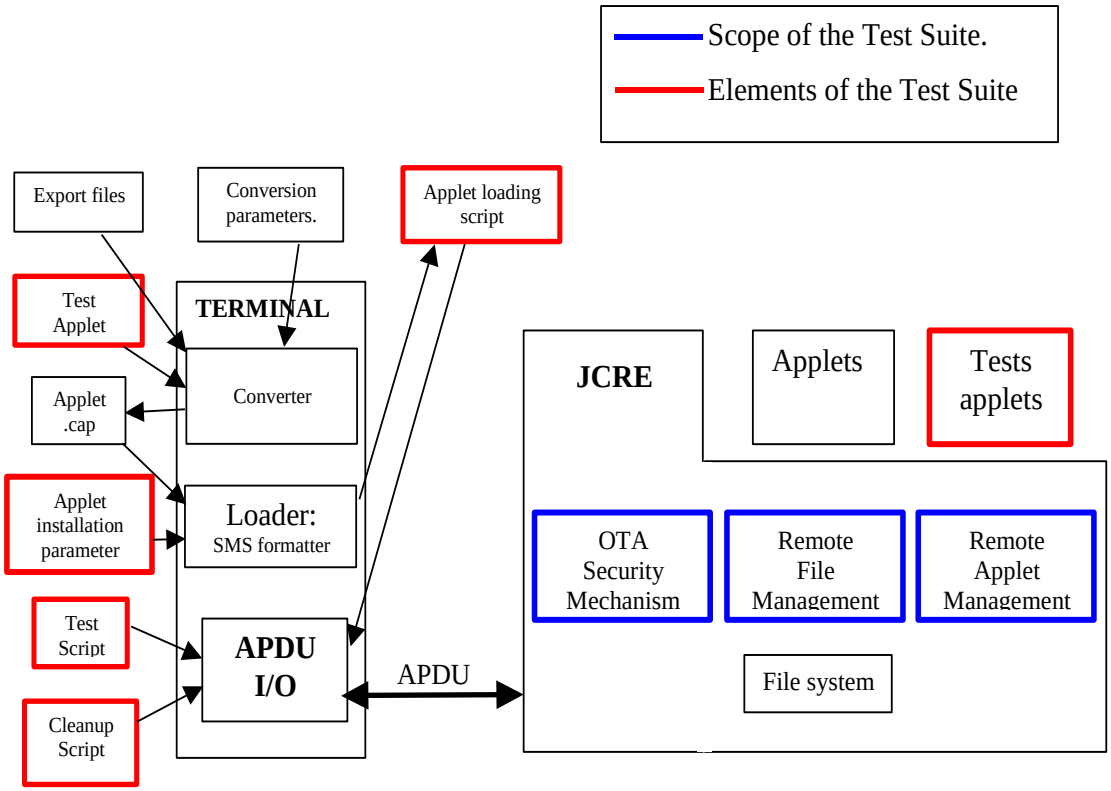
This specification contains tests that would ensure service interoperability between smart cards for "Security Mechanisms for the USAT".

Tests using RFM with no security level are only applicable to smart cards implementing no security level to the RFM application. These tests are listed in the normative Annex G.

Tests using RAM are only applicable to smart cards implementing a minimum security level set to CC integrity . These tests are listed in the normative Annex G.

4.2 Test environment description

The general architecture for the test environment is:



Note: This diagram shows the test architecture required to test interoperability at both API and bytcode level. The latter is currently not included in the current specification. The diagram is for information.

4.3 Tests format

4.3.1 Test Area Reference

4.3.1.1 Format description

The area reference shall be derived from main area, sub area and subject from the 3GPP TS 23.048 [6] as follows:

MainArea1	Description of MainArea1		
	SubArea1	Description of SubArea1	
		Subject1	Description of Subject1
		Subject2	Description of Subject2
	SubArea2	Description of SubArea2	
		Subject1	Description of Subject1

Based on this format description, the test area reference name shall be:

<MainArea>_<SubArea>_<Subject>

See annex A for the acronym table.

4.3.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Normal execution:
  - Contains normal execution, each referenced as a Conformance Requirement Reference Normal (CRRN)
- Error case:
  - Contains error cases, each referenced as a Conformance Requirement Reference Error (CRRE)

4.3.1.2 Test Area Files

The files included in the Test Area use the following naming convention:

- Test Script: [Mode]_[Test Area Reference]_[Test script number].scr
- Test Applet: [Test Area Reference]_[Test applet number].java
- Load Script: [Mode]_[Test Area Reference]_[Load Script number].ldr
- Cleanup Script: [Mode]_[Test Area Reference]_[Cleanup Script number].clr
- Parameter File: [Test Area Reference]_[Parameter File number].par

The field [Mode] takes the values SIM or USIM depending on the type of application, SIM or USIM, for which the test script is dedicated.

The test script, applet, installation parameters, load script, cleanup script and conversion parameters numbers start from '1'.

The test script, load script and cleanup script shall share a common syntax and format (see Annex B).

The parameter file has an own syntax (see Annex G) and contains parameters to be used for CAP-file conversion and loading/cleanup script generation.

Scripts file shall be run in the following order:

[Mode]_ [Test Area Reference]_1.ldr

[Mode]_ [Test Area Reference]_1.scr

[Mode]_ [Test Area Reference]_1.clr

[Mode]_ [Test Area Reference]_2.ldr

[Mode]_ [Test Area Reference]_2.scr

[Mode]_ [Test Area Reference]_2.clr

….

[Mode]_ [Test Area Reference]_n.ldr

[Mode]_ [Test Area Reference]_n.scr

[Mode]_ [Test Area Reference]_n.clr

In case that one of the files is not needed, it shall be skipped during the tests execution.

4.3.1.3 Test Procedure

Each test procedure contains a table to indicate the test description and the expected responses from the applet and/or the APDU level as follows:

Id	Description	Applet Expectation	SIM APDU Expectation	USIM APDU Expectation
	Test Case detailed description	*Applet expected behavior.*	*Expected response at APDU level for a SIM application.*	*Expected response at APDU level for an USIM application.*

4.3.1.4 Test Coverage

The table at the end of each test procedure indicates the correspondence between the Conformance Requirements Reference (CRR) and the different test cases.

4.4 Initial Conditions

The Initial Conditions are a set of general prerequisites for the (U)SIM prior to the execution of testing. For each test procedure described in this document, the following rules apply to the Initial Conditions:

- unless otherwise stated, the file system and the files content shall fulfil the requirements described in the "Default Prepersonalisation" paragraph;
- unless otherwise stated, before installing the applet(s) relevant to the current test procedure, no packages specific to other test procedures shall be present.

When both statements apply, a test procedure is said to be in the "Default Initial Conditions" state.

4.4.1 Security parameters

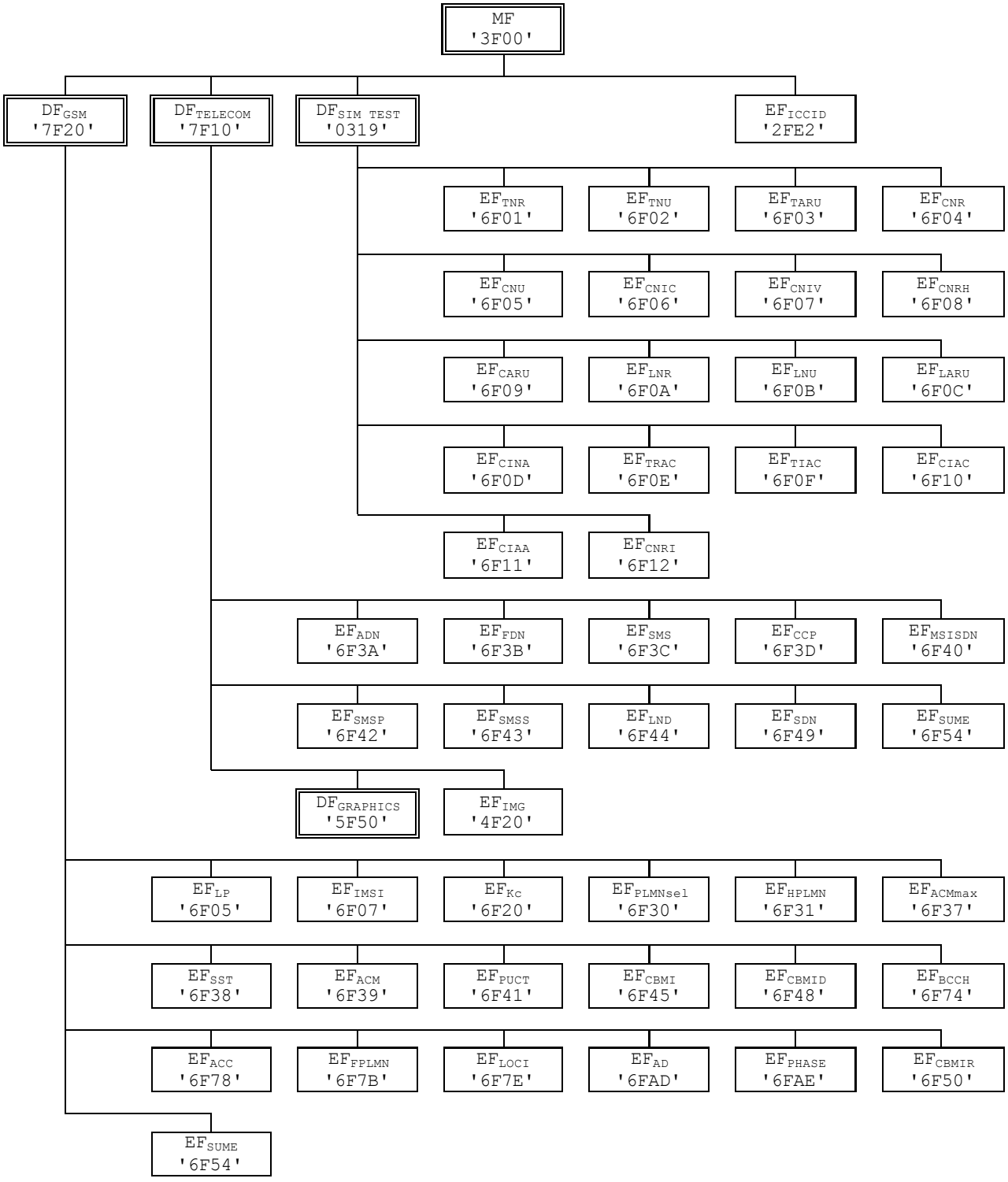
The following key sets are required to run the security tests:

Key set number	Parameter	Value	Comment
1	KIC	01 23 45 67 89 AB CD EF	Keyset 1 is reserved for applet installation and shall not be modified by test cases
	KID	01 23 45 67 89 AB CD EF	
	Counter	00 00 00 00 00	
	Algo	DES in CBC	
2	KIC	01 23 45 67 89 AB CD EF	Used for Security tests on SIM SMS PP
	KID	01 23 45 67 89 AB CD EF	

	Counter	00 00 00 00 00	
	Algo	DES in CBC	
3	KIC	01 23 45 67 89 AB CD EF	Used for Security tests on USIM SMS PP
	KID	01 23 45 67 89 AB CD EF	
	Counter	00 00 00 00 00	
	Algo	DES in CBC	
4	KIC	01 23 45 67 89 AB CD EF	Used for Security tests on SIM SMS PP Response Packet
	KID	01 23 45 67 89 AB CD EF	
	Counter	00 00 00 00 00	
	Algo	DES in CBC	
5	KIC	01 23 45 67 89 AB CD EF	Used for Security tests on USIM SMS PP Response Packet
	KID	01 23 45 67 89 AB CD EF	
	Counter	00 00 00 00 00	
	Algo	DES in CBC	
6	KIC	01 23 45 67 89 AB CD EF	Used for Security tests on SIM SMS CB
	KID	01 23 45 67 89 AB CD EF	
	Counter	00 00 00 00 00	
	Algo	DES in CBC	
7	KIC	01 23 45 67 89 AB CD EF	Used for Security tests on USIM SMS CB
	KID	01 23 45 67 89 AB CD EF	
	Counter	00 00 00 00 00	
	Algo	DES in CBC	
9	KIC	01 23 01 23 01 23 01 23 32 10 32 10 32 10 32 10	Used for Security tests on SIM and USIM SMS PP and CB
	KID	32 10 32 10 32 10 32 10 01 23 01 23 01 23 01 23	
	Counter	00 00 00 00 00	
	Algo	Triple DES in outer-CBC mode using two different keys	
10	KIC	11 11 11 11 11 11 11 11 22 22 22 22 22 22 22 22 33 33 33 33 33 33 33 33	Used for Security tests on SIM and USIM SMS PP and CB
	KID	01 01 01 01 01 01 01 01 02 02 02 02 02 02 02 02 03 03 03 03 03 03 03 03	
	Counter	00 00 00 00 00	
	Algo	Triple DES in outer-CBC mode using three different keys	
15	KIC	AA AA AA AA AA AA AA AA	Used for Security tests on SIM and USIM SMS PP and CB
	KID	EE EE EE EE EE EE EE EE	
	Counter	00 00 00 00 00	
	Algo	DES in ECB mode	

4.4.2 Prepersonalisation

The following table presentes the minimun prepersonnalisation required to run the test suites.



See annex C for the files description.

4.4.3 Environment

For tests interoperability reason, the Remote File Management Application TAR shall be set to ’01 23 45’ when running in a SIM mode, and to ’01 23 47’ when running in an USIM mode.

4.5 Package name

For the specific tests of 3GPP TS 43.019 [5] compliant cards, the Java packages integrating this Test Suite shall follow this naming convention:

sim.test.security.[Test Area Reference]: Java Card packages containing Test Area for security features

sim.test.remote.[Test Area Reference]: Java Card packages containing Test Area for remote management features

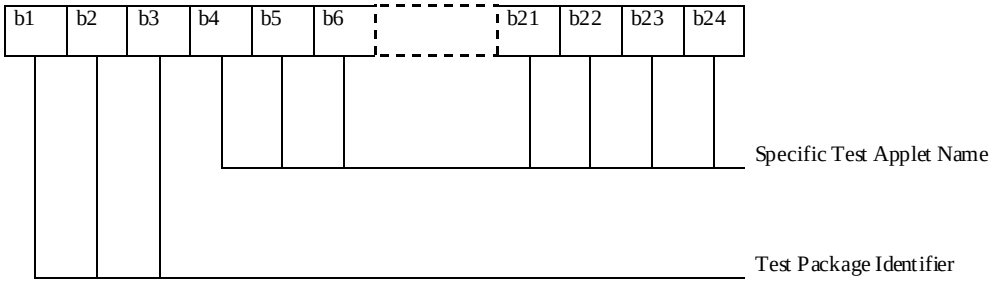
Example: The package *../sim.test.remote.ANA_...* creates the following directory structure

*../sim/test/remote/ANA_.../ANA_..._[1..n].**, where '*ANA_..._[1..n].**' are the different test applets Java source files used in *[Test Area Reference] ANA_....*

4.6 AID Coding

The AID coding for the Test Packages, Applet classes and Applet shall be as specified in 3GPP TS 101 220 [8]. In addition, the following TAR values are defined for use within the present document:

TAR Coding (3 bytes/ 24 bits):



4.6.1 Test package Identifier(bits b1-b3):

000 reserved (as TAR= '00.00.00' is reserved for Card Manager)

001 Test suite

111 sim.test.util

other values are RFU

Application Provider specific data (1 byte):

'00' for Package

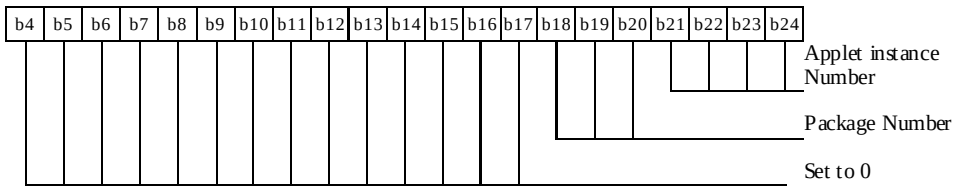
'01' for Applet class

'02' for Applet Instance

For example, the AID of Package sim.test.util is 'A0 00 00 00 09 00 02 FF FF FF FF 89 E0 00 00 00'

4.6.2 Specific Test Applet Name

Specific applet test name (bits b4-b24):



Package number (3 bits): it shall start with 1 for the class and shall be 0 for the package.

Applet Instance number (4 bits) defined in the test procedure it shall start with 1 for the applet instance and shall be 0 for package and class.

4.7 Test Equipment

These subclauses recommend a minimum specification for each of the items of test equipment referenced in the tests.

4.7.1 APDU tool

This test tool shall meet the following requirements:

- be able to send command to the card TPDU;
- be able to check none, only a part, or all of the data returned;
- be able to check none, only part, or all of the status returned;
- be able to accept all valid status codes returned;
- be able to support Reader commands;
- if there is an error in data or status returned, the tool shall return an error.

4.7.2 Util packages

Annex D includes loading , testing and cleaning script examples.

4.7.3 Applet installation parameters

4.7.3.1 Security parameters

Loading scripts shall use the following security parameters as stated in 3GPP TS 23.048 [6] for applet installation:

Parameter	Value in hexadecimal
SPI	0A 00
KIC	00
KID	11
TAR	00 00 00
PCNTR	00

4.7.3.2 Loading components

Cap files in loading scripts shall not include the descriptor component as described in Java Card 2.1 VM Architecture Specification [7].

4.8 Testing methodology

4.8.1 Test interfaces and facilities

The (U)SIM-ME interface provides the main transport interface for the purpose of performing conformance tests.

5 Test specification

5.1 Generalised secured packet structure

5.1.1 Command packet structure

5.1.1.1 Conformance Requirements

Normal execution

CRRN1: The receiving application, indicated by the TAR field, processes the command packet once the security checks have been performed successfully.

CRRN2: The security of a command packet is defined according to SPI first byte and can combine encryption, integrity and anti-replay features.

CRRN3: The bit3 of SPI1 is used with Kic byte to specify which type of encryption is applied to the command packet. The DES (in CBC and ECB modes) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN4: The bits b1b2 bit of SPI1 are used with KID field to specify which type of integrity check protects the command packet. The DES (in CBC mode) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN5: The bits b4b5 of SPI1 are used to specify how should the anti-replay be checked with the CNTR field: CNTR can be either greater or incremented by 1 compared to the last accepted command packet.

CRRN6: The different security features are processed in the following order: The receiving entity first deciphers the secured command packet, then checks its integrity and finally checks the anti-replay counter.

CRRN7: The anti-replay counter of the receiving entity is only updated once all the security checks are performed successfully.

CRRN9: If the SPI1 indicates that no RC, CC or DS is present in the Command Header, the RC/CC/DS field shall be of zero length.

CRRN10: A command packet where SPI1 indicates “no counter available” has its 5 byte CNTR field present.

CRRN11: In case of a ciphered command packet, the PCNTR indicates the number of padding bytes in the Secured Data field which are not processed by the receiving application.

Error cases

CRRE1: The receiving entity does not perform the security verification if the CPI is not a 23.048 [6] secured command packet identifier.

CRRE2: The command packet is discarded if the CHL field is inconsistent.

CRRE3: No data is sent to the receiving application when the receiving entity fails to decipher the message if required.

CRRE4: No data is sent to the receiving application when the RC/CC/DS field check fails.

CRRE5: No data is sent to the receiving application when the CNTR field is lower or equal to the counter of the receiving entity, if b5 of SPI1 is set to 1.

CRRE6: No data is sent to the receiving application when the CNTR field is more than 1 unit greater than the counter of the receiving entity, if b4b5 of SPI1 is 11.

CRRE7: If SPI1 indicates that RC, CC or DS is present in the Command Header and if padding is required, the padding octets shall be coded '00'. These octets shall not be included in the secured data. Otherwise, the message is rejected.

5.1.2 Response packet structure

5.1.2.1 Conformance Requirements

Normal execution

CRRN1: The response packet is sent by the receiving entity when the command packet format is correct and SPI2 requires a PoR, even when a ciphering, integrity or anti-replay error occurs.

CRRN2: The security of a response packet is defined according to the second byte of SPI and can combine encryption and integrity.

CRRN3: If an error occurs in the security checks or in the receiving application and b2b1 of SPI2 is set to 10 (PoR on error), then a response packet is sent back by the receiving entity.

CRRN4: The TAR and CNTR fields of the deciphered response packet are the same as in the deciphered command packet.

CRRN5: The RC/CC/DS field is not included in the response packet when b4b3 in SPI2 are set to 00 (No RC/CC/DS).

CRRN6: The response packet is sent in unciphered when b5 of SPI2 is set to 0.

CRRN7: The bit5 of SPI2 is used with Kic byte to specify which type of encryption is applied to the response packet. The DES (in CBC and ECB modes) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN8: The bits b3b4 bit of SPI2 are used with KID field to specify which type of integrity check protects the response packet. The DES (in CBC mode) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN9: In case of a ciphered response packet, the PCNTR indicates the number of padding bytes appended in the Secured Data field.

CRRN10: If a command packet with a PoR required is successfully delivered to the receiving application, then the response status code in the corresponding response packet is 0 (PoR OK).

Error cases

CRRE1: The receiving entity sends a response packet with a Response Status Code set to '01' (RC/CC/DS failed) if there is an error in the calculation of RC/CC/DS and a PoR is requested.

CRRE2: The receiving entity sends a response packet with a Response Status Code set to '05' (ciphering error) when deciphering fails in a ciphered command packet with PoR requesting encryption . This occurs e.g. when bits b5-b8 of Kic indicate an incorrect key identifier or when the ciphered data length is not correct.

CRRE3: The receiving entity sends a response packet with a Response Status Code set to '02' (CNTR low) when the CNTR field is lower than or equal to the counter of the receiving entity, if bit b5 of SPI1 is set to 1 and a PoR is requested.

CRRE4: The receiving entity sends a response packet with a Response Status Code set to '03' (CNTR high) when the CNTR field is more than 1 unit greater than the counter of the receiving entity, if b4b5 of SPI1 is 11 and a PoR is requested.

CRRE5: The receiving entity sends a response packet with a Response Status Code set to '04' (CNTR blocked) when the counter of the receiving entity is set to its maximum value (0xFFFFFFFFFF), if b5 of SPI1 is 1 and a PoR is requested.

CRPP6: The receiving entity sends a response packet with a Response Status Code set to '09' (TAR unknown) when there no application matched by this TAR, if a PoR is requested.

CRPP7: The receiving entity sends a response packet with a Response Status Code set to '0A' (Insufficient security level) when the application matched by this TAR has a minimum security level higher than the command packet one and a PoR is requested.

5.2 Implementation for SMS-PP

5.2.1 Structure of the SMS

5.2.1.1 Commands Description

Test Area Reference: SEC_SPP_SSS

5.2.1.1.1 Conformance Requirements

Normal execution

CRRN1: The command packet shall be accepted if the SMS-DELIVER, SMS-SUBMIT, SMS-DELIVER-REPORT or SMS-SUBMIT-REPORT header indicates that the data is binary (8 bit).

Error cases

CRRE1: The command packet is discarded if the UDHI bit is not set as defined in 3GPP TS 23.040 [2].

5.2.1.1.2 Test Area Files

Test Applet: n.a.

Load Script: n.a.

Test Script: (U)SIM_SEC_SPP_SSS _1.scr

Cleanup Script: (U)SIM_SEC_SPP_SSS _1.clr.

Parameter File: n.a.

5.2.1.1.3 Test Procedure

Default settings :

SPI:

No counter available

No RC, CC or DS

No ciphering

PoR required to be sent to the SE

No RC, CC or DS applied to PoR response to SE

PoR response shall no be ciphered

PoR shall be sent using SMS-DELIVER-REPORT

KIC:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

KID:

keyset 1

00: Algorithm known implicitly by both entities

00: DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

PCNTR 00

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**Accept SMS with 8 bit binary data** **Good Case : SMS with 8 bit binary data** 1- SD (121 bytes): SELECT MF, SELECT DF_{SIM}			

	TEST, SELECT EF_{TARU}, UPDATE BINARY 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0B 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E with offset 0 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 01010203	2- SW=9000, expected data shall be 01010203
2	**Accept only SMS with UDHI bit set** **Good Case : UDHI bit set** 1- UDHI bit set SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad Case : UDHI bit not set** 3- UDHI bit not set SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed		2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0201	2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0201

5.2.1.1.4 Test Coverage

CRR number	Test case number
N1	1
E1	2

5.2.2 Command Packet contained in a Single SMS-PP

5.2.2.1 Commands Description

Test Area Reference: SEC_SPP_CSS

5.2.2.1.1 Conformance Requirements

Normal execution

CRRN1: In order to include a Command Packet inside a Single SMS-PP, the SMS-DELIVER data structure as defined in 3GPP TS 23.040 [2] is used.

CRRN2: The User Data Header of the SMS-PP is composed of one TLV field with a Tag value of 0x70 and a length value of 0x00 (and, therefore, an empty Value field). This TLV represents the Command Packet Identifier.

CRRN3: All fields from the CPL to the Secured Data (except CHI, which is a Null field) of the Command Packet are stored in order in the SM field of the SMS-PP structure.

CRRN4: The Command Packet Length field is coded over two octets. It shall not be coded according to ISO/IEC 7816-6.

CRRN5: The Command Header Length field is coded over one octet. It shall not be coded according to ISO/IEC 7816-6.

CRRN6: All fields from the SPI to the Secured Data are coded as defined in the Generalised Command Packet Structure.

CRRN7: The Command Packet Length and Command Header Length fields are included in the calculation of the RC/CC/DS, if used.

CCRN8: The maximum length of the user data within one single SMS-PP shall be 140 bytes.

Error cases

CRRE1: The receiving entity does not perform the security verification if the CPI is not a 23.048 [6] secured command packet identifier.

CRRE2: The command packet is discarded if the CHL field is inconsistent.

5.2.2.1.2 Test Area Files

Test Applet: n.a.

Load Script: n.a.

Test Script: SEC_SPP_CSS_1.scr

Cleanup Script: SEC_SPP_CSS_1.clr

Parameter File: n.a.

5.2.2.1.3 Test Procedure

Default settings :

SPI:

No counter available

No RC, CC or DS

No ciphering

PoR required to be sent to the SE

No RC, CC or DS applied to PoR response to SE

PoR response shall no be ciphered

PoR shall be sent using SMS-DELIVER-REPORT

KIC:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

KID:

keyset 1

00: Algorithm known implicitly by both entities

00: DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

PCNTR 00

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**Maximum length of user data is 140 bytes** **Good Case : SMS with 140 bytes user data** 1- SD (121 bytes): SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0B 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E with offset 0 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad Case : SMS with 141 bytes user data** 3- SD (122 bytes): SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 03 02 03 04 05 06 07 08 09 0A 0B 0C 0B 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F with offset 0 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 01010203…5D5E 4- SW=9000, expected data shall be 01010203…5D5E	2- SW=9000, expected data shall be 01010203…5D5E 4- SW=9000, expected data shall be 01010203…5D5E
2	**CPL and CHL are included in the CC calculation** **Good Case : Correct CC calculation** 1- SPI Cryptographic Checksum , SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 with offset 0, CC is calculated with CPL and CHL SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad Case : CPL not used for CC calculation** 3- SPI Cryptographic Checksum , SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 with offset 0, CC is calculated without CPL SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Bad Case : CHL not used for CC calculation** 5- SPI Cryptographic Checksum , SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 with offset 0, CC is calculated without CHL SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0201 6- SW=9000, expected data shall be 0201	2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0201 6- SW=9000, expected data shall be 0201
3	**Incorret value of CPI** 1- CNTR: 00 00 00 00 00 PCNTR: 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 CPI: 00			

	SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0201	2- SW=9000, expected data shall be 0201
4	**inconsistent CHL field** 1- CNTR: 00 00 00 00 00 PCNTR: 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 CHL: 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0201	2- SW=9000, expected data shall be 0201

5.2.2.1.4 Test Coverage

CRR number	Test case number
N1	Tested in (U)SIM_SEC_SPP_SSS
N2	Tested in (U)SIM_SEC_SPP_SSS
N3	Tested in (U)SIM_SEC_SPP_SSS
N4	Tested in (U)SIM_SEC_SPP_SSS
N5	Tested in (U)SIM_SEC_SPP_SSS
N6	Tested in (U)SIM_SEC_SPP_SSS
N7	2
N8	1
E1	3
E2	4

5.2.3 Command Packet contained in a Concatenated SMS-PP

5.2.3.1 Commands Description

Test Area Reference: SEC_SPP_CCS

5.2.3.1.1 Conformance Requirements

Normal execution

CRRN1: If the length of a Command Packet exceeds 140 octets, the Concatenated SMS mechanism as described in 3GPP TS 23.040 [2] shall be used.

CRRN2: The User Data Header of the first SMS consists of:
- The Concatenation Control Header TLV according to 3GPP TS 23.040 [2] (5 octets).
- The Command Packet Identifier as a TLV with Tag value 0x70 and Length value 0x00.

CRRN3: The two elements of the User Data Header (IEIa and IEIb) of the first SMS can be given in any order.

CRRN4: The User Data Header of subsequent SMS consists only of the Concatenated Control Header TLV.

CRRN5: The CPL to RC/CC/DS fields are coded as in a Single SMS-PP for the first SMS, and are not present in all subsequent SMS'.

CRRN6: For the first SMS, the value of the CPL field shall contain the length of the complete Command Packet, including all parts of the Secured Data.

CRRN7: If the data is ciphered, then it is ciphered before being broken down into individual concatenated elements.

CRRN8: The Command Packet Length and Command Header Length fields are included in the calculation of the RC/CC/DS, if used.

5.2.3.1.2 Test Area Files

Test Applet: n.a.

Load Script: n.a.

Test Script: (U)SIM_SEC_SPP_CCS_1.scr

Cleanup Script: (U)SIM_SEC_SPP_CCS_1.clr

Parameter File: n.a.

5.2.3.1.3 Test Procedure

Default settings :

SPI:

No counter available

No RC, CC or DS

No ciphering

PoR required to be sent to the SE

No RC, CC or DS applied to PoR response to SE

PoR response shall no be ciphered

PoR shall be sent using SMS-DELIVER-REPORT

KIC:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

KID:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

PCNTR 00

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**No Ciphering and No Integrity** **Good Case : Send 1st of 2 concatenated SMS, UDH (IEIb, IEIa)** 1- SD – part 1 (121 bytes): SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0B 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B			

	3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E with offset 0 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD not executed **Good Case : Send 2nd of 2 concatenated SMS** 3- SD – part 2: Continue the UPDATE BINARY from the 1st SMS. 5F 60 61 62 63 64 65 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Good Case : Send 1st of 2 concatenated SMS** 5- SD – part 2: UPDATE BINARY 01 07 with offset 0 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed **Good Case : Send 2nd of 2 concatenated SMS** 7- - SD – part 1: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 with offset 0 SMS-PP-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be FFFFFF…FF 4- SW=9000, expected data shall be 01010203…6465 6- SW=9000, expected data shall be 01010203 8- SW=9000, expected data shall be 0107	2- SW=9000, expected data shall be FFFFFF…FF 4- SW=9000, expected data shall be 01010203…6465 6- SW=9000, expected data shall be 01010203 8- SW=9000, expected data shall be 0107
2	**Mixed mode Ciphering and Integrity** **Good Case : Send 1st of 2 concatenated SMS** 1- SPI, Redundancy Check, Ciphering KID, DES, DES in CBC mode KIC, DES, DES in CBC mode SD – part 1 (137 bytes): SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01, UPDATE BINARY 02 02, UPDATE BINARY 02 03, UPDATE BINARY 02 04, UPDATE BINARY 02 05, UPDATE BINARY 02 06, UPDATE BINARY 02 07, UPDATE BINARY 02 08, UPDATE BINARY 02 09, UPDATE BINARY 02 0A, UPDATE BINARY 02 0B, UPDATE BINARY 02 0C, all updates with offset 0 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD not executed **Good Case : Send 2nd of 2 concatenated SMS** 3- SPI, Redundancy Check, Ciphering KID, DES, DES in CBC mode KIC, DES, DES in CBC mode SD – part 2: UPDATE BINARY 02 0D, UPDATE BINARY 02 0E, UPDATE BINARY 02 0F SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0107 4- SW=9000, expected	2- SW=9000, expected data shall be 0107 4- SW=9000, expected

			data shall be 020F	data shall be 020F

5.2.3.1.4 Test Coverage

CRR number	Test case number
N1	1, 2
N2	1, 2
N3	1
N4	1, 2
N5	1, 2
N6	1, 2
N7	2
N8	2

5.2.4 Response packet structure

5.2.4.1 Commands Description

Test Area Reference: SEC_SPP_RPS

5.2.4.1.1 Conformance Requirements

Normal execution

CRRN1: The Single SMS-PP Response Packet is contained in the response message delivered by the UICC through SMS-DELIVER-REPORT or SMS-SUBMIT depending on b6 of SPI2.

CRRN2: The User Data Header of the Single SMS-PP response message is composed of one TLV field with a Tag value of 0x71 and a length value of 0x00.

CRRN3: When a Response Packet is too large to be contained in a Single SMS-PP a Response Packet containing the Status Code "More Time" should be returned followed by a complete Response Packet, which may be concatenated.

CRRN4: All fields of the Response Packet from the RPL to the Additional Response Data (except the RHI which is a Null field) are stored in order in the SM field of the response message structure.

CRRN5: The Response Packet Length field is coded over two octets. It shall not be coded according to ISO/IEC 7816-6.

CRRN6: The Response Header Length field is coded over one octet. It shall not be coded according to ISO/IEC 7816-6.

CRRN7: All fields from the TAR to the RC/CC/DS are coded as defined in the Generalised Response Packet Structure.

CRRN8: The Response Packet Length and the three preceding octets (UDHL and the Tag and Length fields from the UDH) are included in the calculation of the RC/CC/DS, if used.

5.2.4.1.2 Test Area Files

Test Applet: n.a.

Load Script: n.a.

Test Script: n.a.

Cleanup Script: n.a.

Parameter File: n.a.

5.2.4.1.3 Test Procedure

N.a. because tested within SEC_SPP_SMR.

5.2.4.1.4 Test Coverage

CRR number	Test case number
N1	Tested within (U)SIM_SEC_SPP_SMR
N2	Tested within (U)SIM_SEC_SPP_SMR
N3	Not testable
N4	Tested within (U)SIM_SEC_SPP_SMR
N5	Tested within (U)SIM_SEC_SPP_SMR
N6	Tested within (U)SIM_SEC_SPP_SMR
N7	Tested within (U)SIM_SEC_SPP_SMR
N8	Tested within (U)SIM_SEC_SPP_SMR

5.2.5 Security Mechanism for the Command Packet

5.2.5.1 Commands Description

Test Area Reference: SEC_SPP_SMC

5.2.5.1.1 Conformance Requirements

Normal execution

CRRN1: The receiving application, indicated by the TAR field, processes the command packet once the security checks have been performed successfully.

CRRN2: The security of a command packet is defined according to SPI first byte and can combine encryption, integrity and anti-replay features.

CRRN3: The bit3 of SPI1 is used with Kic byte to specify which type of encryption is applied to the command packet. The DES (in CBC and ECB modes) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN4: The bits b1b2 of SPI1 are used with KID field to specify which type of integrity check protects the command packet. The DES (in CBC mode) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN5: The bits b4b5 of SPI1 are used to specify how should the anti-replay be checked with the CNTR field: CNTR can be either greater or incremented by 1 compared to the last accepted command packet.

CRRN6: The different security features are processed in the following order: The receiving entity first deciphers the secured command packet, then checks its integrity and finally checks the anti-replay counter.

CRRN7: The anti-replay counter of the receiving entity is only updated once all the security checks are performed successfully.

CRRN8: If the SPI1 indicates that no RC, CC or DS is present in the Command Header, the RC/CC/DS field shall be of zero length.

CRRN9: A command packet where SPI1 indicates “no counter available” has its 5 bytes CNTR field present.

CRRN10: In case of a ciphered command packet, the PCNTR indicates the number of padding bytes in the Secured Data field which are not processed by the receiving application.

Error cases

CRRE1: No data is sent to the receiving application when the receiving entity fails to decipher the message if required.

CRRE2: No data is sent to the receiving application when the RC/CC/DS field check fails.

CRRE3: No data is sent to the receiving application when the CNTR field is lower or equal to the counter of the receiving entity, if b5 of SPI1 is set to 1.

CRRE4: No data is sent to the receiving application when the CNTR field is more than 1 unit greater than the counter of the receiving entity, if b4b5 of SPI1 is 11.

CRRE5: If SPI1 indicates that RC, CC or DS is present in the Command Header and if padding is required, the padding octets shall be coded '00'. These octets shall not be included in the secured data. Otherwise, the message is rejected.

5.2.5.1.2 Test Area Files

Test Applet: n.a.

Load Script: n.a.

Test Script: (U)SIM_SEC_SPP_SMC_1.scr

(U)SIM_SEC_SPP_SMC_2.scr

(U)SIM_SEC_SPP_SMC_3.scr

(U)SIM_SEC_SPP_SMC_4.scr

Cleanup Script: (U)SIM_SEC_SPP_SMC_1.clr

(U)SIM_SEC_SPP_SMC_2.clr

(U)SIM_SEC_SPP_SMC_3.clr

(U)SIM_SEC_SPP_SMC_4.clr

Parameter File: n.a.

5.2.5.1.3 Test Procedure

5.2.5.1.3.1 (U)SIM_*SEC_SPP_SMC_1*, Testfocus counter

Testfocus: Counter

SPI

00: No counter available (note 1)

01: Counter available; no replay or sequence checking (note 2)

10: Process if and only if counter value is higher than the value in the RE (note 3)

11: Process if and only if counter value is one higher than the value in the RE (note 4)

Default settings:

SPI:

No RC, CC or DS

No cipehring

No PoR required to be sent to the SE

KIC:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

KID:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

PCNTR 00

Counter in Smartcard is 00 00 00 00 00

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**No counter available** **Good case: use maximum counter value** 1- CNTR: FF FF FF FF FF SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Good case: use minimum counter value** 3- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 03 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Bad case: counter missing in CP** 5- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed		2- SW=9000, expected data shall be 0101 4- SW=9000, expected data shall be 0103 6- SW=9000, expected data shall be 0103	2- SW=9000, expected data shall be 0101 4- SW=9000, expected data shall be 0103 6- SW=9000, expected data shall be 0103
2	**Counter available; no replay or sequence checking** **Good case : use maximum counter value** 1- CNTR: FF FF FF FF FF SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Good case: use minimum counter value** 3- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Bad case : counter missing in CP** 5- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed		2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0203 6- SW=9000, expected data shall be 0203	2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0203 6- SW=9000, expected data shall be 0203
3	**Process if and only if counter value is higher than the value in the RE** **Good case : counter one higher then in the RE** 1- CNTR: 00 00 00 00 01 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01 SMS-PP-DOWNLOAD			

	2- READ BINARY EF_{TARU}, verify SD executed **Good case : counter 0x10 higher then in RE** 3- CNTR: 00 00 00 00 11 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 03 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Bad case : counter 0x11 lower then in the RE** 5- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 05 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed **Bad case : counter one lower then in the RE** 7- CNTR: 00 00 00 00 10 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 07 SMS-PP-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD not executed **Bad case : counter equal to value in the RE** 9- CNTR: 00 00 00 00 11 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 09 SMS-PP-DOWNLOAD 10- READ BINARY EF_{TARU}, verify SD not executed **Good case : counter 0x0F higher then in the RE** 11- CNTR: 00 00 00 00 20 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 11 SMS-PP-DOWNLOAD 12- READ BINARY EF_{TARU}, verify SD executed **Bad case : counter missing in CP** 13- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 13 SMS-PP-DOWNLOAD 14- READ BINARY EF_{TARU}, verify SD not executed		2- SW=9000, expected data shall be 0301 4- SW=9000, expected data shall be 0303 6- SW=9000, expected data shall be 0303 8- SW=9000, expected data shall be 0303 10- SW=9000, expected data shall be 0303 12- SW=9000, expected data shall be 0311 14- SW=9000, expected data shall be 0311	2- SW=9000, expected data shall be 0301 4- SW=9000, expected data shall be 0303 6- SW=9000, expected data shall be 0303 8- SW=9000, expected data shall be 0303 10- SW=9000, expected data shall be 0303 12- SW=9000, expected data shall be 0311 14- SW=9000, expected data shall be 0311
4	**Process if and only if counter value is one higher than the value in the RE** **Good case : counter one higher then in the RE** 1- CNTR: 00 00 00 00 21 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 01 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case : counter 0x02 higher then in RE** 3- CNTR: 00 00 00 00 23 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 03 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Bad case : counter 0x21 lower then in the RE** 5- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT		2- SW=9000, expected data shall be 0401 4- SW=9000, expected data shall be 0401	2- SW=9000, expected data shall be 0401 4- SW=9000, expected data shall be 0401

	EF_{TARU}, UPDATE BINARY 04 05 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed **Bad case : counter one lower then in the RE** 7- CNTR: 00 00 00 00 20 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 07 SMS-PP-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD not executed **Bad case : counter equal to value in the RE** 9- CNTR: 00 00 00 00 21 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 09 SMS-PP-DOWNLOAD 10- READ BINARY EF_{TARU}, verify SD not executed **Good case : counter one higher then in the RE** 11- CNTR: 00 00 00 00 22 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 11 SMS-PP-DOWNLOAD 12- READ BINARY EF_{TARU}, verify SD executed **Bad case : counter missing in CP** 13- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 13 SMS-PP-DOWNLOAD 14- READ BINARY EF_{TARU}, verify SD not executed		6- SW=9000, expected data shall be 0401 8- SW=9000, expected data shall be 0401 10- SW=9000, expected data shall be 0401 12- SW=9000, expected data shall be 0411 14- SW=9000, expected data shall be 0411	6- SW=9000, expected data shall be 0401 8- SW=9000, expected data shall be 0401 10- SW=9000, expected data shall be 0401 12- SW=9000, expected data shall be 0411 14- SW=9000, expected data shall be 0411

5.2.5.1.3.2 (U)SIM_SEC_SPP_SMC_2, Testfocus integrity

Testfocus : Integrity

SPI

01: Redundancy Check

10: Cryptographic Checksum

KID

01: DES

00: DES in CBC mode

01: Triple DES in outer-CBC mode using two different keys

10: Triple DES in outer-CBC mode using three different keys

Default settings

SPI:

No cipehring

No PoR response to SE

KIC:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

Keysets :

Keyset 2 (SIM), keyset 3 (USIM), key for DES in CBC mode

Keyset 9, key for Triple DES in outer-CBC mode using two different keys

Keyset 10, key for Triple DES in outer-CBC mode using three different keys

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

PCNTR 00

Length of RC/CC equal to 8

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**Redundancy Check** 1. Not testable, because no mandatory algorithm specified.			
2	**Cryptographic Checksum, DES in CBC mode** **Good case: correct CC** 1- KID : Keyset 2 (SIM), keyset 3 (USIM), DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01, SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect CC** 3- KID : Keyset 2 (SIM), keyset 3 (USIM), DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03, Toggle bit 0 of the last CC byte after CC calculation SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct CC** 5- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05, SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0201 6- SW=9000, expected data shall be 0205	2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 0201 6- SW=9000, expected data shall be 0205
3	**Cryptographic Checksum, Triple DES in outer-CBC mode using two different keys** **Good case: correct CC** 1- KID : Keyset 9, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01, SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect CC**		2- SW=9000, expected data shall be 0301	2- SW=9000, expected data shall be 0301

	3- KID : Keyset 9, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 03, Toggle bit 0 of the last CC byte after CC calculation SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct CC** 5- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 05, SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		4- SW=9000, expected data shall be 0301 6- SW=9000, expected data shall be 0305	4- SW=9000, expected data shall be 0301 6- SW=9000, expected data shall be 0305
4	**Cryptographic Checksum, Triple DES in outer-CBC mode using three different keys** **Good case: correct CC** 1- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 01, SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect CC** 3- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 03, Toggle bit 0 of the last CC byte after CC calculation SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct CC** 5- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 05, SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0401 4- SW=9000, expected data shall be 0401 6- SW=9000, expected data shall be 0405	2- SW=9000, expected data shall be 0401 4- SW=9000, expected data shall be 0401 6- SW=9000, expected data shall be 0405

5.2.5.1.3.3 (U)SIM_SEC_SPP_SMC_3, Testfocus ciphering

Testfocus : Ciphering

KIC

01: DES

00: DES in CBC mode

01: Triple DES in outer-CBC mode using two different keys

10: Triple DES in outer-CBC mode using three different keys

11: DES in ECB mode

Default settings:

SPI:

No RC, CC or DS

Ciphering

No PoR required to be sent to the SE

KID:

keyset 2 (SIM), keyset 3 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

KIC

01: DES

Keysets:

Keyset 2 (SIM), keyset 3 (USIM), key for DES in CBC mode

Keyset 9, key for Triple DES in outer-CBC mode using two different keys

Keyset 10, key for Triple DES in outer-CBC mode using three different keys

Keyset 15, key for DES in ECB mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**DES in CBC mode** **Good case: correct ciphering** 1- KIC : Keyset 2 (SIM), keyset 3 (USIM) PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering** not testable **Good case: correct padding bytes shall not be processed.** 3- KIC : Keyset 2 (SIM), keyset 3 (USIM) PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 01 05 01 05 01 05 01, padding bytes UPDATE BINARY 01 06 Cipher with 7 padding bytes. SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering** 5- KIC : Keyset 2 (SIM), keyset 3 (USIM) PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 07, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0101 4- SW=9000, expected data shall be 010501050105010501 6- SW=9000, expected data shall be 0107	2- SW=9000, expected data shall be 0101 4- SW=9000, expected data shall be 010501050105010501 6- SW=9000, expected data shall be 0107
2	**Triple DES in outer-CBC mode using two different keys** **Good case: correct ciphering** 1- KIC : Keyset 9 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT			

	EF_{TARU}, UPDATE BINARY 02 01, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering** not testable **Good case: correct ciphering, padding bytes shall not be processed.** 3- KIC : Keyset 9 PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 02 05 02 05 02 05 02, padding bytes UPDATE BINARY 02 06 Cipher with 7 padding bytes. SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Good case: correct ciphering** 5- KIC : Keyset 9 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 07, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 020502050205020502 6- SW=9000, expected data shall be 0207	2- SW=9000, expected data shall be 0201 4- SW=9000, expected data shall be 020502050205020502 6- SW=9000, expected data shall be 0207
3	**Triple DES in outer-CBC mode using three different keys** **Good case: correct ciphering** 1- KIC : Keyset 10 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering** not testable **Good case: correct ciphering, padding bytes shall not be processed.** 3- KIC : Keyset 10 PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 05 03 05 03 05 03 05 03, padding bytes UPDATE BINARY 03 06 Cipher with 7 padding bytes. SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering** 5- KIC : Keyset 10 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 07, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0301 4- SW=9000, expected data shall be 030503050305030503 6- SW=9000, expected data shall be 0307	2- SW=9000, expected data shall be 0301 4- SW=9000, expected data shall be 030503050305030503 6- SW=9000, expected data shall be 0307
4	**DES in EBC mode** **Good case: correct ciphering** 1- KIC : Keyset 15 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 01, Padding bytes			

	00 00 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering** not testable **Good case: correct ciphering, padding bytes shall not be processed.** 3- KIC : Keyset 15 PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 05 04 05 04 05 04 05 04, padding bytes UPDATE BINARY 04 06 Cipher with 7 padding bytes. SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering** 5- KIC : Keyset 15 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 07, Padding bytes 00 00 00 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0401 4- SW=9000, expected data shall be 040504050405040504 6- SW=9000, expected data shall be 0407	2- SW=9000, expected data shall be 0401 4- SW=9000, expected data shall be 040504050405040504 6- SW=9000, expected data shall be 0407

5.2.5.1.3.4 (U)SIM_SEC_SPP_SMC_4, Testfocus mixed mode integrity, ciphering and counter

Default setting

SPI:

10: Cryptographic Checksum

Ciphering

No PoR required to be sent to the SE

KID:

keyset 2 (SIM), keyset 3 (USIM)

01: DES

00: DES in CBC mode

KIC

keyset 2 (SIM), keyset 3 (USIM)

01: DES

00: DES in CBC mode

Keysets:

Keyset 2 (SIM), keyset 3 (USIM), key for DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**Mixed mode Ciphering, Integrity, no counter available** **Good case: correct ciphering and correct CC** 1- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 Padding bytes 00 00 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering (as a result incorrect CC)** 3- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 03 padding bytes 00 00 00 Toggle bit 0 of the last padding byte after correct ciphering SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering and correct CC, padding bytes shall not be processed.** 5- PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 01 05 01 05 01 05 01 padding bytes UPDATE BINARY 01 06 Cipher with 7 padding bytes. SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed **Bad case: correct ciphering and incorrect CC** 7- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 padding bytes 00 00 00 Toggle bit 0 of last CC byte after CC calculation. SMS-PP-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering and correct CC** 9- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 09 Padding bytes 00 00 00 SMS-PP-DOWNLOAD 10- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0101 4- SW=9000, expected data shall be 0101 6- SW=9000, expected data shall be 010501050105010501 8- SW=9000, expected data shall be 0105 10- SW=9000, expected data shall be 0109	2- SW=9000, expected data shall be 0101 4- SW=9000, expected data shall be 0101 6- SW=9000, expected data shall be 010501050105010501 8- SW=9000, expected data shall be 0105 10- SW=9000, expected data shall be 0109
2	**Mixed mode Ciphering, Integrity, Process if and only if counter value is higher than the value in the RE** **Good case: correct ciphering, correct CC, correct counter** 1- CNTR: 00 00 01 00 00 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 Padding bytes 00 00 00 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, expected data shall be 0201	2- SW=9000, expected data shall be 0201

	Bad case:correct ciphering, correct CC, counter to low 3- CNTR: 00 00 01 00 00 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 padding bytes 00 00 00 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000, expected data shall be 0201	4- SW=9000, expected data shall be 0201
	Good case: correct ciphering, correct CC, correct counter 5- CNTR: 00 00 01 00 01 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 Padding bytes 00 00 00 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000, expected data shall be 0205	6- SW=9000, expected data shall be 0205
	Bad case: correct ciphering, incorrect CC, correct counter 7- CNTR: 00 00 01 00 02 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 07 Padding bytes 00 00 00 Toggle bit 0 of last CC byte after CC calculation. SMS-PP-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD executed		8- SW=9000, expected data shall be 0205 10- SW=9000, expected data shall be 0209	8- SW=9000, expected data shall be 0205 10- SW=9000, expected data shall be 0209
	Good case: correct ciphering, correct CC, correct counter 9- CNTR: 00 00 01 00 02 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 09 Padding bytes 00 00 00 SMS-PP-DOWNLOAD 10- READ BINARY EF_{TARU}, verify SD executed		12- SW=9000, expected data shall be 0209	12- SW=9000, expected data shall be 0209
	Bad case: incorrect ciphering (as a result incorrect CC), correct counter 11- CNTR: 00 00 01 00 03 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 11 padding bytes 00 00 00 Toggle bit 0 of the last padding byte after correct ciphering SMS-PP-DOWNLOAD 12- READ BINARY EF_{TARU}, verify SD not executed		14- SW=9000, expected data shall be 0213	14- SW=9000, expected data shall be 0213
	Good case: correct ciphering, correct CC, correct counter 13- CNTR: 00 00 01 00 03 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 13 padding bytes 00 00 00			

	SMS-PP-DOWNLOAD 14- READ BINARY EF_{TARU}, verify SD not executed			

5.2.5.1.4 Test Coverage

CRR number	Test case number
N1	(U)SIM_SEC_SPP_SMC_1: 1, 2, 3, 4 (U)SIM_SEC_SPP_SMC_2: 1, 2, 3, 4 (U)SIM_SEC_SPP_SMC_3: 1, 2, 3, 4 (U)SIM_SEC_SPP_SMC_4: 1, 2
N2	(U)SIM_SEC_SPP_SMC_1: 1, 2, 3, 4 (U)SIM_SEC_SPP_SMC_2: 1, 2, 3, 4 (U)SIM_SEC_SPP_SMC_3: 1, 2, 3, 4 (U)SIM_SEC_SPP_SMC_4: 1, 2
N3	(U)SIM_SEC_SPP_SMC_3: 1, 2, 3, 4
N4	(U)SIM_SEC_SPP_SMC_2: 1, 2, 3, 4
N5	(U)SIM_SEC_SPP_SMC_1: 1, 2, 3, 4
N6	(U)SIM_SEC_SPP_SMC_4: 1, 2
N7	(U)SIM_SEC_SPP_SMC_4: 2
N8	(U)SIM_SEC_SPP_SMC_1: 1, 2, 3, 4
N9	(U)SIM_SEC_SPP_SMC_1: 1
N10	(U)SIM_SEC_SPP_SMC_3: 1, 2, 3, 4
E1	(U)SIM_SEC_SPP_SMR: 1
E2	(U)SIM_SEC_SPP_SMR: 1
E3	(U)SIM_SEC_SPP_SMR: 1
E4	(U)SIM_SEC_SPP_SMC_1:4
E5	(U)SIM_SEC_SPP_SMC_2:2

5.2.6 Security Mechanism for the Response Packet

5.2.6.1 Commands Description

Test Area Reference: SEC_SPP_SMR

5.2.6.1.1 Conformance Requirements

Normal execution

CRRN1: The response packet is sent by the receiving entity when the command packet format is correct and SPI2 requires a PoR, even when a ciphering, integrity or anti-replay error occurs.

CRRN2: The security of a response packet is defined according to the second byte of SPI and can combine encryption and integrity.

CRRN3: If an error occurs in the security checks or in the receiving application and b2b1 of SPI2 is set to 10 (PoR on error), then a response packet is sent back by the receiving entity.

CRRN4: The TAR and CNTR fields of the deciphered response packet are the same as in the deciphered command packet.

CRRN5: The RC/CC/DS field is not included in the response packet when b4b3 in SPI2 are set to 00 (No RC/CC/DS).

CRRN6: The response packet is sent unciphered when b5 of SPI2 is set to 0.

CRRN7: The bit5 of SPI2 is used with Kic byte to specify which type of encryption is applied to the response packet. The DES (in CBC and ECB modes) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN8: The bits b3b4 of SPI2 are used with KID field to specify which type of integrity check protects the response packet. The DES (in CBC mode) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN9: In case of a ciphered response packet, the PCNTR indicates the number of padding bytes appended in the Secured Data field.

CRRN10: If a command packet with a PoR required is successfully delivered to the receiving application, then the response status code in the corresponding response packet is 0 (PoR OK).

Error cases

CRRE1: The receiving entity sends a response packet with a Response Status Code set to '01' (RC/CC/DS failed) if there is an error in the calculation of RC/CC/DS and a PoR is requested.

CRRE2: The receiving entity sends a response packet with a Response Status Code set to '05' (ciphering error) when deciphering fails in a ciphered command packet with PoR requesting encryption. This occurs e.g. when bits b5-b8 of Kic indicate an incorrect key identifier or when the ciphered data length is not correct.

CRRE3: The receiving entity sends a response packet with a Response Status Code set to '02' (CNTR low) when the CNTR field is lower than or equal to the counter of the receiving entity, if bit b5 of SPI1 is set to 1 and a PoR is requested.

CRRE4: The receiving entity sends a response packet with a Response Status Code set to '03' (CNTR high) when the CNTR field is more than 1 unit greater than the counter of the receiving entity, if b4b5 of SPI1 is 11 and a PoR is requested.

CRRE5: The receiving entity sends a response packet with a Response Status Code set to '04' (CNTR blocked) when the counter of the receiving entity is set to its maximum value (0xFFFFFFFFFF), if b5 of SPI1 is 1 and a PoR is requested.

CRRE6: The receiving entity sends a response packet with a Response Status Code set to '09' (TAR unknown) when there is no application matched by this TAR, if a PoR is requested.

CRRE7: The receiving entity sends a response packet with a Response Status Code set to '0A' (Insufficient security level) when the application matched by this TAR has a minimum security level higher than the command packet one and a PoR is requested.

5.2.6.1.2 Test Area Files

Test Applet: n.a.

Load Script: n.a.

Test Script: (U)SIM_SEC_SPP_SMR_1.scr

Cleanup Script: (U)SIM_SEC_SPP_SMR_1.clr

Parameter File: n.a.

5.2.6.1.3 Test Procedure

Testfocus : PoR

SPI

00: No PoR reply to the Sending Entity (SE)

01: PoR required to be sent to the SE

10: PoR required only when an error has occured

00: No RC, CC or DS applied to PoR response to SE

01: PoR response with simple RC applied to it

10: PoR response with CC applied to it

0 : PoR response shall not be ciphered

1 : PoR response shall be ciphered

0 : PoR response shall be sent using SMS-DELIVER-REPORT

1 : PoR response shall be sent using SMS-SUBMIT

Default settings:

SPI:

Process if and only if counter value is higher than the value in the RE

Cryptographic Checksum

Ciphering

KIC:

keyset 4 (SIM), keyset 5 (USIM)

00: Algorithm known implicitly by both entities

00: DES in CBC mode

KID

keyset 4 (SIM), keyset 5 (USIM)

01: DES

00: DES in CBC mode

PCNTR 00

Length of RC/CC equal to 8

Keysets:

keyset 4 (SIM), keyset 5 (USIM), key for DES in CBC mode

Keyset 9, key for Triple DES in outer-CBC mode using two different keys

Keyset 10, key for Triple DES in outer-CBC mode using three different keys

Keyset 15, key for DES in ECB mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

Test procedure

Id	Description	API-Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**PoR required to be sent to the SE** **(**No RC, CC or DS applied to PoR response to SE**)** **(**PoR response shall not be ciphered**)** **Good case: SMS-DELIVER-REPORT** 1- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 01 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 SMS-PP-DOWNLOAD 2- GET RESPONSE 3- READ BINARY EF_{TARU}, verify SD executed **Bad case: SMS-DELIVER-REPORT, CNTR low** 4- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 04 SMS-PP-DOWNLOAD 5- GET RESPONSE 6- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS-DELIVER-REPORT, CC failed** 7- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 01 01, erroneous CC SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 07 SMS-PP-DOWNLOAD 8- GET RESPONSE 9- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS-DELIVER-REPORT, Chipering error** not testable **Good case: SMS-SUBMIT** 10- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 01 10 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 13, SMS-PP-DOWNLOAD 11- FETCH 12- TERMINAL RESPONSE 13- READ BINARY EF_{TARU}, verify SD executed **Bad case: SMS- SUBMIT, CNTR low** 14- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 17, SMS-PP-DOWNLOAD		1- SW=9F13 2- SC=00 ARD=049000 3- SW=9000, expected data shall be 0101 4- SW=9E10 5- SC=02 6- SW=9000, expected data shall be 0101 7- SW=9E10 8- SC=01 9- SW=9000, expected data shall be 0101 10- SW=912D or 91 2F (depending on the Alpha Id TLV) 11- SC=00 ARD=049000 13- SW=9000, expected data shall be 0113 14- SW=912A or 91 2C (depending on the Alpha Id TLV)	1- SW=6113 2- SC=00 ARD=049000 3- SW=9000, expected data shall be 0101 4- SW=6110 5- SC=02 6- SW=9000, expected data shall be 0101 7- SW=6110 8- SC=01 9- SW=9000, expected data shall be 0101 10- SW=912D or 91 2F (depending on the Alpha Id TLV) 11- SC=00 ARD=049000 13- SW=9000, expected data shall be 0113 14- SW=912A or 91 2C (depending on the Alpha Id TLV)

	15- FETCH 16- TERMINAL RESPONSE 17- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS- SUBMIT, CC failed** 18- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 01 11, erroneous CC SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 21, SMS-PP-DOWNLOAD 19- FETCH 20- TERMINAL RESPONSE 21- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS- SUBMIT, Chipering error** not testable		15- SC=02 17- SW=9000, expected data shall be 0113 18- SW=912A or 91 2C (depending on the Alpha Id TLV) 19- SC=01 21- SW=9000, expected data shall be 0113	15- SC=02 17- SW=9000, expected data shall be 0113 18- SW=912A or 91 2C (depending on the Alpha Id TLV) 19- SC=01 21- SW=9000, expected data shall be 0113
2	**No PoR reply to the Sending Entity (SE)** **(**No RC, CC or DS applied to PoR response to SE**)** **(**PoR response shall not be ciphered**)** **Good case: SMS-DELIVER-REPORT** 1- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 02 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: SMS-DELIVER-REPORT, CNTR low** 3- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 SMS-PP-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS-DELIVER-REPORT, CC failed** 5- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 02 01, erroneous CC SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 SMS-PP-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS-DELIVER-REPORT, Chipering error** not testable **Good case: SMS-SUBMIT** 7- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 02 10 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 09 SMS-PP-DOWNLOAD		1- SW=9000 2- SW=9000, expected data shall be 0201 3- SW=9000 4-SW=9000, expected data shall be 0201 5- SW=9000 6- SW=9000, expected data shall be 0201 7- SW=9000	1- SW=9000 2- SW=9000, expected data shall be 0201 3- SW=9000 4-SW=9000, expected data shall be 0201 5- SW=9000 6- SW=9000, expected data shall be 0201 7- SW=9000

	8- READ BINARY EF_{TARU}, verify SD executed **Bad case: SMS- SUBMIT, CNTR low** 9- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 11 SMS-PP-DOWNLOAD 10- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS- SUBMIT, CC failed** 11- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 02 11, erroneous CC SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 13 SMS-PP-DOWNLOAD 12- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS- SUBMIT, Chipering error** not testable		8- SW=9000, expected data shall be 0209 9- SW=9000 10- SW=9000, expected data shall be 0209 11- SW=9000 12- SW=9000, expected data shall be 0209	8- SW=9000, expected data shall be 0209 9- SW=9000 10- SW=9000, expected data shall be 0209 11- SW=9000 12- SW=9000, expected data shall be 0209
3	**PoR required only when an error has occured (**No RC, CC or DS applied to PoR response to SE**)** **(**PoR response shall not be ciphered**)** **Good case: SMS-DELIVER-REPORT** 1- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 03 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01 SMS-PP-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: SMS-DELIVER-REPORT, CNTR low** 3- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 03, SMS-PP-DOWNLOAD 4- GET RESPONSE 5- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS-DELIVER-REPORT, CC failed** 6- PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 03 01, erroneous CC SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 06, SMS-PP-DOWNLOAD 7- GET RESPONSE 8- READ BINARY EF_{TARU}, verify SD not executed **Bad case: SMS-DELIVER-REPORT, Chipering error** not testable **Good case: SMS-SUBMIT**		1- SW=9000 2- SW=9000, expected data shall be 0301 3- SW=9E10 4- SC=02 5- SW=9000, expected data shall be 0301 6- SW=9E10 7- SC=01 8- SW=9000, expected data shall be 0301	1- SW=9000 2- SW=9000, expected data shall be 0301 3- SW=6110 4- SC=02 5- SW=9000, expected data shall be 0301 6- SW=6110 7- SC=01 8- SW=9000, expected data shall be 0301

	9- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 03 10 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 09 SMS-PP-DOWNLOAD		9- SW=9000	9- SW=9000
	10- READ BINARY EF_{TARU}, verify SD executed		10- SW=9000, expected data shall be 0309	10- SW=9000, expected data shall be 0309
	Bad case: SMS- SUBMIT, CNTR low			
	11- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 11 SMS-PP-DOWNLOAD		11- SW=912A or 91 2C (depending on the Alpha Id TLV)	11- SW=912A or 91 2C (depending on the Alpha Id TLV)
	12- FETCH			
	13- TERMINAL RESPONSE		12-SC=02	12-SC=02
	14- READ BINARY EF_{TARU}, verify SD not executed		14-SW=9000, expected data shall be 0309	14-SW=9000, expected data shall be 0309
	Bad case: SMS- SUBMIT, CC failed			
	15- PoR response shall be sent using SMS-SUBMIT COUNTER 00 00 00 03 11, erroneous CC SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 15 SMS-PP-DOWNLOAD		15- SW=912A or 91 2C (depending on the Alpha Id TLV)	15- SW=912A or 91 2C (depending on the Alpha Id TLV)
	16- FETCH			
	17- TERMINAL RESPONSE		16-SC=01	16-SC=01
	18- READ BINARY EF_{TARU}, verify SD not executed		18-SW=9000, expected data shall be 0309	18-SW=9000, expected data shall be 0309
	Bad case: SMS- SUBMIT, Chipering error			
	not testable			
4	**PoR response with simple RC applied to it** not testable			
5	**PoR response with CC applied to it** **(**PoR required to be sent to the SE**)** **(**PoR response shall not be ciphered**)**			
	Good case: SMS-DELIVER-REPORT			
	1- PoR response shall be sent using SMS-DELIVER-REPORT PoR response with CC applied to it COUNTER 00 00 00 05 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 05 01 SMS-PP-DOWNLOAD		1- SW=9FXX	1- SW=61XX
	2- GET RESPONSE		2- CC=	2- CC=
	3- READ BINARY EF_{TARU}, verify SD executed		3- SW=9000, expected data shall be 0501	3- SW=9000, expected data shall be 0501
	Good case: SMS-SUBMIT			
	4- PoR response shall be sent using SMS-SUBMIT PoR response with CC applied to it COUNTER 00 00 00 05 01 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 05 04, SMS-PP-DOWNLOAD		4- SW= 9135 or 91 37 (depending on the Alpha Id TLV)	4- SW= SW=9135 or 91 37 (depending on the Alpha Id TLV)
	5- FETCH			

	6- TERMINAL RESPONSE 7- READ BINARY EF_{TARU}, verify SD executed		7- SW=9000, expected data shall be 0504	7- SW=9000, expected data shall be 0504
6	**PoR response shall be ciphered** **(**PoR required to be sent to the SE**)** **(**PoR response shall be ciphered**)** **(**No RC, CC or DS applied to PoR response to SE**)** **DES in CBC mode** **Good case: correct ciphering** 1- KIC : Keyset 2 (SIM), keyset 3 (USIM) PCNTR: 01 SD: SELECT MF padding byte 00 PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 06 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 06 01, SMS-PP-DOWNLOAD,SMS-PP-DOWNLOAD 2- GET RESPONSE 3 - READ BINARY EF_{TARU} to verify that SD have been executed **Triple DES in outer-CBC mode using two different keys** **Good case: correct ciphering** 4- KIC : Keyset 9 PCNTR: 01 SD: SELECT MF padding byte 00 PoR response shall be sent using SMS-DELIVER-REPORT SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 06 02, SMS-PP-DOWNLOAD,SMS-PP-DOWNLOAD 5- GET RESPONSE 6 - READ BINARY EF_{TARU} to verify that SD have been executed **Triple DES in outer-CBC mode using three different keys** **Good case: correct ciphering** 7- KIC : Keyset 10 PCNTR: 01 SD: SELECT MF padding byte 00 PoR response shall be sent using SMS-SUBMIT SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 06 03, SMS-PP-DOWNLOAD,SMS-PP-DOWNLOAD 6 - FETCH 7 - TERMINAL RESPONSE 8 - READ BINARY EF_{TARU} to verify that SD have been executed **DES in EBC mode** **Good case: correct ciphering** 9 - KIC : Keyset 15 PCNTR: 01 SD: SELECT MF padding byte 00 PoR response shall be sent using SMS-SUBMIT		1- SW=9F19 3 - SW = 90 00, expected data shall be 06 01 4- SW=9F19 6 - SW = 90 00, expected data shall be 06 02 7 - SW=91 33 or 91 35 (depending on the Alpha Id TLV) 8 - SW = 90 00, expected data shall be 06 03 9 - SW=91 33 or 91 35 (depending on the Alpha Id TLV) 12 - SW = 90 00,	1- SW=6119 3 - SW = 90 00, expected data shall be 06 01 4- SW=6119 6 - SW = 90 00, expected data shall be 06 02 7 - SW=91 33 or 91 35 (depending on the Alpha Id TLV) 8 - SW = 90 00, expected data shall be 06 03 9 - SW=91 33 or 91 35 (depending on the Alpha Id TLV) 12 - SW = 90 00,

	SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 06 04, SMS-PP-DOWNLOAD,SMS-PP-DOWNLOAD 10 - FETCH 11 - TERMINAL RESPONSE 12 - READ BINARY EF_{TARU} to verify that SD have been executed		expected data shall be 06 04	expected data shall be 06 04
7	**PoR response shall be ciphered and CC applied to it** **(**PoR required to be sent to the SE**)** **(**PoR response shall be ciphered**)** **(**PoR response with CC applied to it**)** **DES in CBC mode** **Good case: correct ciphering** 1- KIC : keyset 4 (SIM), keyset 5 (USIM) PCNTR: 01 PoR response shall be sent using SMS-DELIVER-REPORT SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 07 01, SMS-PP-DOWNLOAD,SMS-PP-DOWNLOAD 2- GET RESPONSE 3 - READ BINARY EF_{TARU} to verify that SD have been executed **Triple DES in outer-CBC mode using two different keys** **Good case: correct ciphering** 4- KIC : Keyset 9 PCNTR: 01 SD: SELECT MF padding byte 00 PoR response shall be sent using SMS-DELIVER-REPORT SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 07 02, SMS-PP-DOWNLOAD 5 - GET RESPONSE 6 - READ BINARY EF_{TARU} to verify that SD have been executed **Triple DES in outer-CBC mode using three different keys** **Good case: correct ciphering** 7 - KIC : Keyset 10 PCNTR: 01 SD: SELECT MF padding byte 00 PoR response shall be sent using SMS-SUBMIT SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 07 03, SMS-PP-DOWNLOAD 6- FETCH 7 - TERMINAL RESPONSE 8 - READ BINARY EF_{TARU} to verify that SD have been executed **DES in ECB mode for ciphering and DES in CBC mode for authentication** **Good case: correct ciphering** 8- KIC : Keyset 15 PCNTR: 01 SD: SELECT MF		1- SW=9F21 3- SW=90 00, expected data shall be 07 01 4 – SW = 9F 21 6 - SW = 90 00, expected data shall be 07 02 7 - SW=91 3B or 91 3D (depending on the Alpha Id TLV) 8 - SW = 90 00, expected data shall be 07 03 8- SW=91 3B or 91 3D (depending on the Alpha Id TLV)	1- SW=6121 3- SW=90 00, expected data shall be 07 01 4- SW=6121 6 - SW = 90 00, expected data shall be 07 02 7 - SW=91 3B or 91 3D (depending on the Alpha Id TLV) 8 - SW = 90 00, expected data shall be 07 03 8- SW=91 3B or 91 3D (depending on the Alpha Id TLV)

	padding byte 00 PoR response shall be sent using SMS-SUBMIT SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 07 04, SMS-PP-DOWNLOAD 9- FETCH 10 - TERMINAL RESPONSE 11 - READ BINARY EF_{TARU} to verify that SD have been executed		11 – SW = 90 00, expected data shall be 07 04	11 – SW = 90 00, expected data shall be 07 04
8	**Status Code for CNTR high** **(**PoR required to be sent to the SE**)** (counter only one higher) 1- KIC : keyset 4 (SIM), keyset 5 (USIM) PCNTR: 00 SD: SELECT MF PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 10 00 00 07 00 SMS-PP-DOWNLOAD 2- GET RESPONSE		1- SW=9E10 2- SC=03	1- SW=6110 2- SC=03
9	**Status Code for TAR unknown** **(**PoR required to be sent to the SE**)** 1- KIC : keyset 4 (SIM), keyset 5 (USIM) PCNTR: 00 SD: SELECT MF PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 07 02 TAR: ABCDEF SMS-PP-DOWNLOAD 2- GET RESPONSE		1- SW=9E10 2- SC=09	1- SW=6110 2- SC=09
10	**Status Code for Insufficient security level** **(**PoR required to be sent to the SE**)** 1- KIC : keyset 4 (SIM), keyset 5 (USIM) PCNTR: 00 SD: Install load command PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 00 00 TAR: 000000 SMS-PP-DOWNLOAD 2- GET RESPONSE		1- SW=9E10 2- SC=0A	1- SW=6110 2- SC=0A
11	**Status Code for CNTR blocked** **(**PoR required to be sent to the SE**)** 1- KIC : keyset 4 (SIM), keyset 5 (USIM) PCNTR: 00 SD: SELECT MF PoR response shall be sent using SMS-DELIVER-REPORT COUNTER FF FF FF FF FF SMS-PP-DOWNLOAD 2- KIC : keyset 4 (SIM), keyset 5 (USIM) PCNTR: 00 SD: SELECT MF PoR response shall be sent using SMS-DELIVER-REPORT COUNTER 00 00 00 07 20 SMS-PP-DOWNLOAD 3- GET RESPONSE		2- SW=9E10 3- SC=04	2- SW=6110 3- SC=04

5.2.6.1.4 Test Coverage

CRR number	Test case number
N1	1
N2	3, 4, 5, 6, 7
N3	3
N4	1
N5	1
N6	1
N7	6
N8	5
N9	6
N10	1
E1	1
E2	1
E3	1
E4	8
E5	11
E6	9
E7	10

5.3 Implementation for SMS-CB

5.3.1 Structure of the CBS page in the SMS-CB Message

Test area reference : SEC_SCB_SCB

5.3.1.1 Conformance Requirements

The structure of a Cell Broadcast is defined is 3GPP TS 23.041 [10].

Normal execution

CRRN1: The Cell Broadcast Service (CBS) page must be a fixed block of 88 octets.

CRRN2: The CBS page consists of a 6-octet header and 82 user octets.

CRRN3: The 6-octet Header must be unsecured.

CRRN4: The 6-octet Header includes: Serial Number, Message IDentifier, Data Coding Scheme and Page Parameter.

CRRN5: The Serial Number (SN) is coded on 2 octets and contains the identifier of the message.

CRRN6: The Message Identifier (MID) is coded on 2 octets and contains the source and the type of the message. The value of the MID must be from '1080' to '109F' for command packet CBS messages secured according to 3GPP TS 23.048 [6].

CRRN7: Data Coding Scheme (DCS) is coded on 1 octet and contains the alphabet coding and language.

CRRN8: The Page Parameter (PP) is coded on 2 octets, the Most Significant Nibble contains the page number and the Less Significant Nibble contains the total number of pages.

Error cases

CRRE1: If the CHL field is inconsistent the message shall be discarded.

CRRE2: No data is sent to the receiving application if the Message Identifier (MID) is not in the range '1080' to '109F'.

5.3.1.2 Test suites files

Test Script: (U)SIM_ SEC_SCB_SCB _1.scr

(U)SIM_ SEC_SCB_SCB _1.clr

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**The CBS page consist of a fixed block of 88 bytes** 1- Send a formatted SMS-CB with no security which contains the following commands: - Select Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 01 01 01 2 - Select EF_{TARU} 3 - Read binary command.		1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 01 01 01	1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 01 01 01
2	**A CBS page with inconsistent CHL shall not be processed** 1 – Send a formatted SMS-CB with no security with an inconsistent CHL containing the following commands: - Select Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 02 02 02 2 - Select EF_{TARU} 3 - Read binary command.		1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 01 01 01	1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 01 01 01
3	**The command packet identifier shall be in the range of '1080' to '109F'** 1 - Send a formatted SMS-CB with no security with a command identifier field set to '107F' , containing the following commands: - Select Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 02 02 02 2 - Select EF_{TARU} 3 - Read binary command. 4 - Send a formatted SMS-CB with no security with a command identifier field set to '10A0' , containing the following commands: - Select Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 02 02 02 5 - Select EF_{TARU} 6 - Read binary command. 7 - Send a formatted SMS-CB with no security with a command identifier field set to '1080' , containing the following commands: - Select Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 03 02 07 8 - Select EF_{TARU} 9 - Read binary command. 10 - Send a formatted SMS-CB with no security with a command identifier field set to '109F' , containing the following commands: - Select Select $DF_{SIMTEST}$. - Select EF_{TARU}.		1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 01 01 01 4 - SW= 90 00 6 - SW = 90 00 , expected data shall be 01 01 01 7 - SW= 90 00 9 - SW = 90 00 , expected data shall be 03 02 07 10 - SW= 90 00	1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 01 01 01 4 - SW= 90 00 6 - SW = 90 00 , expected data shall be 01 01 01 7 - SW= 90 00 9 - SW = 90 00 , expected data shall be 03 02 07 10 - SW= 90 00

	- Update Binary command with data 03 02 10 11 - Select EF_{TARU} 12 - Read binary command.		12 - SW = 90 00 , expected data shall be 03 02 10	12 - SW = 90 00 , expected data shall be 03 02 10

5.3.1.3 Test coverage

CRR number	Test case number
CRRN1,…8	1
CRRE1	2
CRRE2	3

5.3.2 A Command Packet structure contained in a SMS-CB message

Test area reference : SEC_SCB_CCB

5.3.2.1 Conformance Requirements

Normal execution

CRRN1: The CPI is coded on 2 octets, these octets are the MID octets.

CRRN2: All fields from the CPL to the Secured Data (composed of the CPL, CHI (Null field), CHL, SPI to RC/CC/DS) of the Command Packet are stored in the order defined in the generalised secured packet structure.

CRRN3: The Command Packet Length (CPL) field is coded over two octets. It shall not be coded according to ISO/IEC 7816-6.

CRRN4: The Command Header Length (CHL) field is coded over one octet. It shall not be coded according to ISO/IEC 7816-6.

CRRN5: All fields from the SPI to the Secured Data are coded as defined in the Generalised Command Packet Structure.

CRRN6: The Command Packet Length and Command Header Length fields are included in the calcu lation of the RC/CC/DS, if used.

CRRN7: In case of several pages, the first CBS page includes the 6-octet Header and the Command Header and the following pages only include the 6-octet Header.

5.3.2.2 Test suites files

Test Script: (U)SIM_ SEC_SCB_CCB _1.scr

(U)SIM_ SEC_SCB_CCB _1.clr

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**CPL and CHL shall be included in the calculation of the RC/CC/DS** **Good case : CPL and CHL are included in CC calculation** 1- Send a SMS-CB message using integrity containing the following command - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 02 03 01 2 - Select EF_{TARU} 3 - Read binary command. **Bad case : CPL and CHL are not included in CC calculation** 4- Send a SMS-CB message using integrity containing the following command - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 02 03 04 5 - Select EF_{TARU} 6 - Read binary command.		1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 02 03 01 4 - SW= 90 00 6 - SW = 90 00 , expected data shall be 02 03 01	1 - SW= 90 00 3 - SW = 90 00 , expected data shall be 02 03 01 4 - SW= 90 00 6 - SW = 90 00 , expected data shall be 02 03 01
2	**The first page contains the 6 octets header and the command header** 1- Send a formatted SMS-CB message composed of two pages containing the following command: - Select $DF_{SIMTEST}$ - Select EF_{TARU} Update binary command with data 00 01 02 …28. 2 - Select EF_{TARU} 3 - Read binary command		1 - SW = 90 00 3 - SW = 90 00 , expected data shall be 00 01 02 …28	1 - SW = 90 00 3 - SW = 90 00 , expected data shall be 00 01 02 …28

5.3.2.3 Test coverage

CRR number	Test case number
CRRN1,2,3,4,5	Tested in other parts of the specification
CRRN6	1
CRRN7	2

5.3.3 Security mechanism for SMS-CB

Test area reference : SEC_SCB_SMC

5.3.3.1 Conformance Requirements

Normal execution

CRRN1: The receiving application, indicated by the TAR field, processes the command packet once the security checks have been performed successfully.

CRRN2: The security of a command packet is defined according to SPI first byte and can combine encryption, integrity and anti-replay features.

CRRN3: The bit3 of SPI1 is used with Kic byte to specify which type of encryption is applied to the command packet. The DES (in CBC and ECB modes) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN4: The bits b1b2 of SPI1 are used with KID field to specify which type of integrity check protects the command packet. The DES (in CBC mode) and TDES algorithms (with 2 or 3 keys in outer-CBC mode) can be used.

CRRN5: The bits b4b5 of SPI1 are used to specify how should the anti-replay be checked with the CNTR field: CNTR can be either greater or incremented by 1 compared to the last accepted command packet.

CRRN6: The different security features are processed in the following order: The receiving entity first deciphers the secured command packet, then checks its integrity and finally checks the anti-replay counter.

CRRN7: The anti-replay counter of the receiving entity is only updated once all the security checks are performed successfully.

CRRN8: If the SPI1 indicates that no RC, CC or DS is present in the Command Header, the RC/CC/DS field shall be of zero length.

CRRN9: A command packet where SPI1 indicates “no counter available” has its 5 bytes CNTR field present.

CRRN10: In case of a ciphered command packet, the PCNTR indicates the number of padding bytes in the Secured Data field which are not processed by the receiving application.

Error cases

CRRE1: No data is sent to the receiving application when the receiving entity fails to decipher the message if required.

CRRE2: No data is sent to the receiving application when the RC/CC/DS field check fails.

CRRE3: No data is sent to the receiving application when the CNTR field is lower or equal to the counter of the receiving entity, if b5 of SPI1 is set to 1.

CRRE4: No data is sent to the receiving application when the CNTR field is more than 1 unit greater than the counter of the receiving entity, if b4b5 of SPI1 is 11.

CRRE5: If SPI1 indicates that RC, CC or DS is present in the Command Header and if padding is required, the padding octets shall be coded '00'. These octets shall not be included in the secured data. Otherwise, the message is rejected.

5.3.3.2 Test suites files

Test Script: n.a.

Test Applet: n.a.

Load Script: n.a.

Test Script: (U)SIM_SEC_SCB_SMC_1.scr

(U)SIM_SEC_SCB_SMC_2.scr

(U)SIM_SEC_SCB_SMC_3.scr

(U)SIM_SEC_SCB_SMC_4.scr

Cleanup Script: (U)SIM_SEC_SCB_SMC_1.clr

(U)SIM_SEC_SCB_SMC_2.clr

(U)SIM_SEC_SCB_SMC_3.clr

(U)SIM_SEC_SCB_SMC_4.clr

Parameter File: n.a.

5.3.3.3 Test procedure

5.3.3.3.1 (U)SIM_*SEC_SCB_SMC_1*, Testfocus counter

Testfocus: Counter

SPI

- 00: No counter available (note 1)
- 01: Counter available; no replay or sequence checking (note 2)
- 10: Process if and only if counter value is higher than the value in the RE (note 3)
- 11: Process if and only if counter value is one higher than the value in the RE (note 4)

Default settings:

SPI:

No RC, CC or DS

No ciphering

No PoR required

KIC:

keyset 6 (SIM), keyset 7 (USIM)

b2b1 = 00: Algorithm known implicitly by both entities

b4b3 = 00: DES in CBC mode

KID:

keyset 6 (SIM), keyset 7 (USIM)

b2b1 = 00: Algorithm known implicitly by both entities

b4b3 = 00: DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

PCNTR 00

Counter in Smartcard is 00 00 00 00 00

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**No counter available** **Good case: use maximum counter value** 1- CNTR: FF FF FF FF FF SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 SMS-CB-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed		1- SW=9000 2- SW=9000, Expected data shall be 0101	1- SW=9000 2- SW=9000, Expected data shall be 0101

	Good case: use minimum counter value 3- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 03 SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Bad case: counter missing in CP** 5- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 SMS-CB-DOWNLOAD 6- READ BINARY EFTARU, verify SD not executed		3- SW = 9000 4- SW=9000, Expected data shall be 0103 5- SW=9000 6- SW=9000, Expected data shall be 0103	3- SW = 9000 4- SW=9000, Expected data shall be 0103 5- SW=9000 6- SW=9000, Expected data shall be 0103
2	**Counter available; no replay or sequence checking** **Good case : use maximum counter value** 1- CNTR: FF FF FF FF FF SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 SMS-CB-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Good case: use minimum counter value** 3- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD executed **Bad case : counter missing in CP** 5- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 SMS-CB-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed		1-SW=9000 2- SW=9000, Expected data shall be 0201 3-SW=9000 4- SW=9000, Expected data shall be 0203 5-SW=9000 6- SW=9000, Expected data shall be 0203	1-SW=9000 2- SW=9000, Expected data shall be 0201 3-SW=9000 4- SW=9000, Expected data shall be 0203 5-SW=9000 6- SW=9000, Expected data shall be 0203
3	**Process if and only if counter value is higher than the value in the RE** **Good case : counter one higher than in the RE** 1- CNTR: 00 00 00 00 01 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01 SMS-CB-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Good case : counter 0x10 higher than in RE** 3- CNTR: 00 00 00 00 11 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 03 SMS-CB-DOWNLOAD		1- SW=9000 2- SW=9000, Expected data shall be 0301 3-SW=9000	1-SW=9000 2- SW=9000, Expected data shall be 0301 3-SW=9000

	4- READ BINARY EF_{TARU}, verify SD executed		4- SW=9000, Expected data shall be 0303	4- SW=9000, Expected data shall be 0303
	Bad case : counter 0x11 lower than in the RE			
	5- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 05 SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD not executed		6- SW=9000, Expected data shall be 0303	6- SW=9000, Expected data shall be 0303
	Bad case : counter one lower than in the RE			
	7- CNTR: 00 00 00 00 10 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 07 SMS-CB-DOWNLOAD		7-SW=9000	7-SW=9000
	8- READ BINARY EF_{TARU}, verify SD not executed		8- SW=9000, Expected data shall be 0303	8- SW=9000, Expected data shall be 0303
	Bad case : counter equal to value in the RE			
	9- CNTR: 00 00 00 00 11 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 09 SMS-CB-DOWNLOAD		9-SW=9000	9-SW=9000
	10- READ BINARY EF_{TARU}, verify SD not executed		10- SW=9000 Expected data shall be 0303	10- SW=9000 Expected data shall be 0303
	Good case : counter 0x0F higher than in the RE			
	11- CNTR: 00 00 00 00 20 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 11 SMS-CB-DOWNLOAD		11-SW=9000	11-SW=9000
	12- READ BINARY EF_{TARU}, verify SD executed		12- SW=9000 Expected data shall be 0311	12- SW=9000 Expected data shall be 0311
	Bad case : counter missing in CP			
	13- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 13 SMS-CB-DOWNLOAD		13-SW=9000	13-SW=9000
	14- READ BINARY EF_{TARU}, verify SD not executed		14- SW=9000, Expected data shall be 0311	14- SW=9000 Expected data shall be 0311
4	**Process if and only if counter value is one higher than the value in the RE**			
	Good case : counter one higher than in the RE			
	1- CNTR: 00 00 00 00 21 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 01 SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000, Expected data shall be 0401	2- SW=9000, Expected data shall be 0401
	Bad case : counter 0x02 higher than in RE			
	3- CNTR: 00 00 00 00 23 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 03 SMS-CB-DOWNLOAD		3-SW=9000	3-SW=9000

	4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000, Expected data shall be 0401	4- SW=9000, Expected data shall be 0401
	Bad case : counter 0x21 lower than in the RE			
	5- CNTR: 00 00 00 00 00 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 05 SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD not executed		6- SW=9000, Expected data shall be 0401	6- SW=9000, Expected data shall be 0401
	Bad case : counter one lower than in the RE			
	7- CNTR: 00 00 00 00 20 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 07 SMS-CB-DOWNLOAD		7-SW=9000	7-SW=9000
	8- READ BINARY EF_{TARU}, verify SD not executed		8- SW=9000, Expected data shall be 0401	8- SW=9000, Expected data shall be 0401
	Bad case : counter equal to value in the RE			
	9- CNTR: 00 00 00 00 21 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 09 SMS-CB-DOWNLOAD		9-SW=9000	9-SW=9000
	10- READ BINARY EF_{TARU}, verify SD not executed		10- SW=9000, Expected data shall be 0401	10- SW=9000, Expected data shall be 0401
	Good case : counter one higher than in the RE			
	11- CNTR: 00 00 00 00 22 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 11 SMS-CB-DOWNLOAD		11-SW=9000	11-SW=9000
	12- READ BINARY EF_{TARU}, verify SD executed		12- SW=9000, Expected data shall be 0411	12- SW=9000, Expected data shall be 0411
	Bad case : counter missing in CP			
	13- remove CNTR from CP SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 13 SMS-CB-DOWNLOAD		13-SW=9000	13-SW=9000
	14- READ BINARY EF_{TARU}, verify SD not executed		14- SW=9000, Expected data shall be 0411	14- SW=9000, Expected data shall be 0411

5.3.3.3.2 (U)SIM_SEC_SCB_SMC_2, Testfocus integrity

Testfocus: Integrity

SPI

01: Redundancy Check

10: Cryptographic Checksum

KID

01: DES

00: DES in CBC mode

01: Triple DES in outer-CBC mode using two different keys

10: Triple DES in outer-CBC mode using three different keys

Default settings:

SPI:

No ciphering

No PoR response to SE

KIC:

keyset 6 (SIM), keyset 7 (USIM)

b2b1 = 00: Algorithm known implicitly by both entities

b4b3 = 00: DES in CBC mode

KID:

keyset 6 (SIM), keyset 7 (USIM) for DES in CBC mode

keyset 9, key for Triple DES in outer-CBC mode using two different keys

keyset 10, key for Triple DES in outer-CBC mode using three different keys

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

PCNTR 00

Length of RC/CC equal to 8

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**Redundancy Check** Not testable, because no mandatory algorithm specified.			
2	**Cryptographic Checksum, DES in CBC mode** **Good case: correct CC** 1- KID : Keyset 6(SIM),Keyset7(USIM), DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01, SMS-CB-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect CC** 3- KID : Keyset 6(SIM),Keyset7(USIM), DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03, Toggle bit 0 of the last CC byte after CC calculation SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct CC**		 1-SW=9000 2- SW=9000, Expected data shall be 0201 3-SW=9000 4- SW=9000, Expected data shall be 0201	 1-SW=9000 2- SW=9000, Expected data shall be 0201 3-SW=9000 4- SW=9000, Expected data shall be 0201

	5- KID : Keyset 6(SIM),Keyset7(USIM), DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05, SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000, Expected data shall be 0205	6- SW=9000, Expected data shall be 0205
3	**Cryptographic Checksum, Triple DES in outer-CBC mode using two different keys**			
	Good case: correct CC			
	1- KID : Keyset 9, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01, SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000,Expected data shall be 0301	2- SW=9000,Expected data shall be 0301
	Bad case: incorrect CC			
	3- KID : Keyset 9, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 03, Toggle bit 0 of the last CC byte after CC calculation SMS-CB-DOWNLOAD		3-SW=9000	3-SW=9000
	4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000,Expected data shall 0301	4- SW=9000,Expected data shall 0301
	Good case: correct CC			
	5- KID : Keyset 9, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 05, SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000 ,Expected data shall be 0305	6- SW=9000 ,Expected data shall be 0305
4	**Cryptographic Checksum, Triple DES in outer-CBC mode using three different keys**			
	Good case: correct CC			
	1- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 01, SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000,Expected data shall be 0401	2- SW=9000,Expected data shall be 0401
	Bad case: incorrect CC			
	3- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 03, Toggle bit 0 of the last CC byte after CC calculation SMS-CB-DOWNLOAD		3-SW=9000	3-SW=9000
	4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000,Expected data shall be 0401	4- SW=9000,Expected data shall be 0401
	Good case: correct CC			
	5- KID : Keyset 10, DES SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 05, SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000

	6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000 ,Expected data shall be 0405	6- SW=9000 ,Expected data shall be 0405

5.3.3.3.3 (U)SIM_SEC_SCB_SMC_3, Testfocus ciphering

Testfocus: Ciphering

KIC

01: DES

00: DES in CBC mode

01: Triple DES in outer-CBC mode using two different keys

10: Triple DES in outer-CBC mode using three different keys

11: DES in ECB mode

Default settings:

SPI:

No RC, CC or DS

Ciphering

No PoR required to be sent to the SE

KID:

keyset 6 (SIM), keyset 7 (USIM)

b2b1 = 00: Algorithm known implicitly by both entities

b4b3 = 00: DES in CBC mode

KIC

b2b1 = 01: DES

Keysets:

keyset 6 (USIM), keyset 7 (USIM) keys for DES in CBC mode

keyset 9, key for Triple DES in outer-CBC mode using two different keys

keyset 10, key for Triple DES in outer-CBC mode using three different keys

keyset 15, key for DES in ECB mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**DES in CBC mode**			

	Good case: correct ciphering 1- KIC : Keyset 6(SIM),KeySet7(USIM) PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000,Expected data shall be 010501050105010501	2- SW=9000,Expected data shall be 010501050105010501
	Bad case: incorrect ciphering not testable			
	Good case: correct ciphering, padding bytes shall not be processed3- KIC : Keyset 6(SIM),KeySet7(USIM) PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 01 05 01 05 01 05 01, padding bytes UPDATE BINARY 01 06 Cipher with 7 padding bytes. SMS-CB-DOWNLOAD		3-SW=9000	3-SW=9000
	4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000,Expected data shall be 0101	4- SW=9000,Expected data shall be 0101
	Good case: correct ciphering 5- KIC : Keyset 6(SIM),KeySet7(USIM) PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 07, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000,Expected data shall be 0107	6- SW=9000,Expected data shall be 0107
2	**Triple DES in outer-CBC mode using two different keys**			
	Good case: correct ciphering 1- KIC : Keyset 9 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000,Expected data shall be 020502050205020502	2- SW=9000,Expected data shall be 020502050205020502
	Bad case: incorrect ciphering not testable			
	Good case: correct ciphering, padding bytes shall not be processed 3- KIC : Keyset 9 PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 02 05 02 05 02 05 02, padding bytes UPDATE BINARY 02 06		3-SW=9000	3-SW=9000

	Cipher with 7 padding bytes. SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000,Expected data shall be 0201	4- SW=9000,Expected data shall be 0201
	Good case: correct ciphering 5- KIC : Keyset 9 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 07, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000,Expected data shall be 0207	6- SW=9000,Expected data shall be 0207
3	**Triple DES in outer-CBC mode using three different keys**			
	Good case: correct ciphering 1- KIC : Keyset 10 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 01, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000,Expected data shall be 030503050305030503	2- SW=9000,Expected data shall be 030503050305030503
	Bad case: incorrect ciphering not testable			
	Good case: correct ciphering, padding bytes shall not be processed 3- KIC : Keyset 10 PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 05 03 05 03 05 03 05 03, padding bytes UPDATE BINARY 03 06 Cipher with 7 padding bytes. SMS-CB-DOWNLOAD		3-SW=9000	3-SW=9000
	4- READ BINARY EF_{TARU}, verify SD not executed		4- SW=9000,Expected data shall be 0301	4- SW=9000,Expected data shall be 0301
	Good case: correct ciphering 5- KIC : Keyset 10 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 03 07, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		5-SW=9000	5-SW=9000
	6- READ BINARY EF_{TARU}, verify SD executed		6- SW=9000,Expected data shall be 0307	6- SW=9000,Expected data shall be 0307
4	**DES in EBC mode**			
	Good case: correct ciphering 1- KIC : Keyset 15 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 01, Padding bytes 00 00 00 SMS-CB-DOWNLOAD		1-SW=9000	1-SW=9000
	2- READ BINARY EF_{TARU}, verify SD executed		2- SW=9000,Expected	2- SW=9000,Expected

	Bad case: incorrect ciphering not testable **Good case: correct ciphering, padding bytes shall not be processed** 3- KIC : Keyset 15 PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 05 04 05 04 05 04 05 04, padding bytes UPDATE BINARY 02 06 Cipher with 7 padding bytes. SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering** 5- KIC : Keyset 15 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 04 07, Padding bytes 00 00 00 SMS-CB-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed		data shall be 040504050405040504 3-SW=9000 4- SW=9000,Expected data shall be 0401 5-SW=9000 6- SW=9000,Expected data shall be 0407	data shall be 040504050405040504 3-SW=9000 4- SW=9000,Expected data shall be 0401 5-SW=9000 6- SW=9000,Expected data shall be 0407

5.3.3.3.4 (U)SIM_SEC_SCB_SMC_4, Testfocus mixed mode integrity, ciphering and counter

Default setting

SPI:

Cryptographic checksum

Ciphering

No PoR required to be sent to the SE

KID:

keyset 6 (SIM), keyset 7 (USIM)

b2b1 = 01: DES

b4b3 = 00: DES in CBC mode

KIC

keyset 6 (SIM), keyset 7 (USIM)

b2b1 = 01: DES

b4b3 = 00: DES in CBC mode

Keysets:

keyset 6 (SIM), keyset 7 (USIM), key for DES in CBC mode

TAR_{SIM} 01 23 45

TAR_{USIM} 01 23 47

CNTR 00 00 00 00 00

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}			
1	**Mixed mode Ciphering, Integrity, no counter available** **Good case: correct ciphering and correct CC** 1- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 01 Padding bytes 00 00 00 SMS-CB-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering (as a result incorrect CC)** 3- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 03 padding bytes 00 00 00 Toggle bit 0 of the last padding byte after correct ciphering SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering and correct CC, padding bytes shall not be processed.** 5- PCNTR: 07 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 01 05 01 05 01 05 01 padding bytes UPDATE BINARY 01 06 Cipher with 7 padding bytes. SMS-CB-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD not executed **Bad case: correct ciphering and incorrect CC** 7- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 05 padding bytes 00 00 00 Toggle bit 0 of last CC byte after CC calculation. SMS-CB-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering and correct CC** 9- PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 01 09 Padding bytes 00 00 00 SMS-CB-DOWNLOAD		1-SW=9000 2- SW=9000,Expected data 0101 3- SW=9000 4- SW=9000,Expected data shall be 0101 5-SW=9000 6- SW=9000,Expected data 010501050105010501 7-SW=9000 8- SW=9000,Expected data 0105 9-SW=9000 10- SW=9000,Expected	1-SW=9000 2- SW=9000,Expected data 0101 3- SW=9000 4- SW=9000,Expected data shall be 0101 5-SW=9000 6- SW=9000,Expected data 010501050105010501 7-SW=9000 8- SW=9000,Expected data 0105 9-SW=9000 10- SW=9000,Expected

	10- READ BINARY EF_{TARU}, verify SD executed		data 0109	data 0109
2	**Mixed mode Ciphering, Integrity, Process if and only if counter value is higher than the value in the RE** **Good case: correct ciphering, correct CC, correct counter** 1- CNTR: 00 00 01 00 00 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 01 Padding bytes 00 00 00 SMS-CB-DOWNLOAD 2- READ BINARY EF_{TARU}, verify SD executed **Bad case:correct ciphering, correct CC, counter too low** 3- CNTR: 00 00 01 00 00 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 03 padding bytes 00 00 00 SMS-CB-DOWNLOAD 4- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering, correct CC, correct counter** 5- CNTR: 00 00 01 00 01 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 05 Padding bytes 00 00 00 SMS-CB-DOWNLOAD 6- READ BINARY EF_{TARU}, verify SD executed **Bad case: correct ciphering, incorrect CC, correct counter** 7- CNTR: 00 00 01 00 02 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 07 Padding bytes 00 00 00 Toggle bit 0 of last CC byte after CC calculation. SMS-CB-DOWNLOAD 8- READ BINARY EF_{TARU}, verify SD executed **Good case: correct ciphering, correct CC, correct counter** 9- CNTR: 00 00 01 00 02 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 09 Padding bytes 00 00 00		1-SW=9000 2- SW=9000,Expected data shall be 0201 3-SW=9000 4- SW=9000,Expected data shall be 0201 5-SW=9000 6- SW=9000, Expected data shall be 0205 7-SW=9000 8- SW=9000, Expected data shall be 0205 9-SW=9000	1-SW=9000 2- SW=9000,Expected data shall be 0201 3-SW=9000 4- SW=9000,Expected data shall be 0201 5-SW=9000 6- SW=9000, Expected data shall be 0205 7-SW=9000 8- SW=9000, Expected data shall be 0205 9-SW=9000

	SMS-CB-DOWNLOAD 10- READ BINARY EF_{TARU}, verify SD executed **Bad case: incorrect ciphering (as a result incorrect CC), correct counter** 11- CNTR: 00 00 01 00 03 PCNTR: 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 02 11 padding bytes 00 00 00 Toggle bit 0 of the last padding byte after correct ciphering SMS-CB-DOWNLOAD 12- READ BINARY EF_{TARU}, verify SD not executed **Good case: correct ciphering, correct CC, correct counter** 13- CNTR: 00 00 01 00 03 SD: SELECT MF, SELECT $DF_{SIM\ TEST}$, SELECT EF_{TARU}, UPDATE BINARY 00 01 02…28 SMS-CB-DOWNLOAD 14- READ BINARY EF_{TARU}, verify SD not executed		10- SW=9000, Expected data shall be 0209 11-SW=9000 12- SW=9000, Expected data 0209 13-SW=9000 14- SW=9000, Expected data shall be 00 01 02…28	10- SW=9000, Expected data shall be 0209 11-SW=9000 12- SW=9000, Expected data 0209 13-SW=9000 14- SW=9000, Expected data shall be 00 01 02…28

5.3.3.4 Test coverage

CRR number	Test case number
N1	(U)SIM_SEC_SCB_SMC_1: 1 , 2, 3, 4 (U)SIM_SEC_SCB_SMC_2: 1, 2, 3, 4 (U)SIM_SEC_SCB_SMC_3: 1, 2, 3, 4 (U)SIM_SEC_SCB_SMC_4: 1, 2
N2	(U)SIM_SEC_SCB_SMC_1: 1, 2, 3, 4 (U)SIM_SEC_SCB_SMC_2: 1, 2, 3, 4 (U)SIM_SEC_SCB_SMC_3: 1, 2, 3, 4 (U)SIM_SEC_SCB_SMC_4: 1, 2
N3	(U)SIM_SEC_SCB_SMC_3: 1, 2, 3, 4
N4	(U)SIM_SEC_SCB_SMC_2: 1, 2, 3, 4
N5	(U)SIM_SEC_SCB_SMC_1: 1, 2, 3, 4
N6	(U)SIM_SEC_SCB_SMC_4: 1, 2
N7	(U)SIM_SEC_SCB_SMC_4: 2
N8	(U)SIM_SEC_SCB_SMC_1: 1,2,3,4
N9	(U)SIM_SEC_SCB_SMC_1: 1, 2, 3, 4
N10	(U)SIM_SEC_SCB_SMC_3: 1
E1	Not testable
E2	(U)SIM_SEC_SCB_SMC_2: 1, 2, 3, 4
E3	(U)SIM_SEC_SCB_SMC_1: 1, 2, 3, 4
E4	(U)SIM_SEC_SCB_SMC_2: 1, 2, 3, 4
E5	(U)SIM_SEC_SCB_SMC_2: 1, 2, 3, 4

5.4 Remote File Management for SIM

5.4.1 Behaviour of the Remote File Management Application

5.4.1.1 Command session description

Test area reference: RFM_SRB_CMDS

5.4.1.1.1 Conformance Requirement

CRRN1: The parameter(s) in the Data Download Message to UICC is either a single command, or a list of commands and shall be processed sequentially.

CRRN2: The application shall take parameters from the Data Download Message to UICC and shall act upon the GSM files according to these parameters.

CRRN3: A Command "session" is defined as starting upon receipt of the parameter/command list, and ends when the parameter list in the Data Download Message to UICC is completed, or when an error is detected which shall halt further processing of the command list.

CRRN4: A command “session” shall not change the logical state (e.g. file pointers) of the UICC.

5.4.1.1.2 Test Area Files

Test Script: SIM_RFM_SRB_CMDS_1.scr

SIM_RFM_SRB_CMDS_1.clr

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**A command session may contain a single command** 1- Send a formatted SMS which contains a Select $DF_{SIMTEST}$ command with a PoR required.		1- SW = 9F 13, additional data expected shall be 01 9F XX	
2	**A command session may contain multiple commands** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 01 01 01		1- SW = 9F 13, additional data expected shall be 03 90 00	
3	**A command session ends when a error occurs** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select Unknow file. - Select EF_{TARU}. - Update Binary command with data 02 02 02 2- Select EF_TARU 3- Read binary command		1- SW = 9F 13, additional data expected shall be 02 94 04 2- SW = 9F XX 3- SW = 90 00, expected data shall be 01 01 01	
4	**A Command session shall not change the logical state of the UICC.** 1- Select $DF_{SIMTEST}$ 2- Select EF_{LARU} 3- Read the selected file using read record command in NEXT access mode. 4- Read the selected file using read record command in NEXT access mode. 5- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Read the selected file using read record command in NEXT access mode 6- Read the selected file using read record command in current access mode		3- SW = 90 00, Returned data shall be 55 55 55 55 4- SW = 90 00, Returned data shall be AA AA AA AA 5- SW =9F 17, additional data expected shall be 03 90 00 55 55 55 55 6- SW = 90 00, Returned data shall be AA AA AA AA	

5.4.1.1.3 Test Coverage

CRR number	Test case number
N1	1, 2
N2	1,2,3,4
N3	3
N4	4

5.4.2 Coding of the command

5.4.2.1 SIM Input command

Test area reference: RFM_SCC_INPT

5.4.2.1.1 Conformance Requirement

CRRN1: The standardised following input commands shall be accepted. The commands are as defined in 3GPP TS 51.011 [3]:

SELECT extended to include the "SELECT by path" command.

UPDATE BINARY

UPDATE RECORD

SEEK

INCREASE

VERIFY CHV

CHANGE CHV

DISABLE CHV

ENABLE CHV

UNBLOCK CHV

INVALIDATE

REHABILITATE

5.4.2.1.2 Test suites files

Test Script: SIM_RFM_SCC_INPT_1.scr

SIM_RFM_SCC_INPT_1.clr

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**A select command shall be accepted** 1- Send a formatted SMS which contains a Select $DF_{SIMTEST}$ command on the applet, with a PoR required.		1- SW = 9F 13, additional data expected shall be 01 9F XX	
2	**A select "by path" command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ / EF_{TARU}.		1- SW = 9F 13, additional data expected shall be 01 9F XX	
3	**An update binary command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{TARU} 3- Update the selected binary command with the following data 00 00 00 4- Send a formatted SMS, with a PoR required in case of error, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 01 01 01 5- Read binary command		3- SW = 90 00 4- SW = 90 00 5- SW = 90 00, expected data shall be 01 01 01	
4	**An update record command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{LARU} 3- Update the selected file with data 00 00 00 00 using update record command in next access mode. 4- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Update the selected file update record command in next access mode with data 01 01 01 01 5- Read the selected file using read record command in current access mode		3- SW = 90 00 4- SW =9F 13, additional data expected shall be 03 90 00 5- SW = 90 00, Returned data shall be 01 01 01 01	
5	**A seek command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Seek command first-forward access mode with data 01 01 01 01		1- SW =9F 13, additional data expected shall be 03 90 00	
6	**An increase command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{CARU} 3- Send a formatted SMS, with a PoR required in case of error, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{CARU} - Increase command with the		3- SW = 90 00	

	following data 01 4- Read the first record of the current selected file by using read record command in absolute access mode.		 4- SW = 90 00, returned data shall be 55 55 56.	
7	**A verify CHV command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following command: - Verify CHV1 with the correct value of the code		 1- SW =9F 13, additional data expected shall be 01 90 00	
8	**A change CHV command shall be accepted** 1- Send a formatted SMS which contains the following command: - Change CHV1 with value 32 32 32 32 2- Send a verify CHV1 command with the new CHV1 value.		 1- SW =90 00 2- SW =90 00	
9	**A disable CHV command shall be accepted** 1- Reset the Card. 2- Select EF_{TIAC} (Invalidate access condition set to CHV1). 3- Send an invalidate command 4- Send a formatted SMS with PoR required which contains the following command: - disable CHV1 command 5- Reset the card 6- Select $DF_{SIMTEST}$ 7- Select EF_{TIAC} 8- Send an invalidate command. 9- Send a rehabilitate EF_{TIAC} command to reset the file context		 2- SW= 9F XX 3- SW =98 04 4- SW= 9F 13, additional data expected shall be 01 90 00 6- SW= 9F XX 7- SW = 9F XX 8- SW = 90 00 9- SW = 90 00	
10	**A enable CHV command shall be accepted** 1- Send a formatted SMS with PoR required which contains the following commands: - enable CHV1 command 2- Reset the card 3- Select $DF_{SIMTEST}$\Select EF_{TIAC} 4- Send an invalidate command.		 1- SW= 9F 13, additional data expected shall be 01 90 00 3- SW = 9F XX4- SW= 98 04	
11	**A unblock CHV command shall be accepted** 1- Block the CHV1 secret code by sending three unsuccessful Verify CHV1 commands. 2- Send a verify CHV1 command with the correct code value. 3- Send a formatted SMS with PoR required which contains the following command: - unblock CHV1 command 4- Send a verify CHV1 command with the correct code value.		 2- SW= 98 40 3- SW= 9F 13, additional data expected shall be 01 90 00 4- SW= 90 00	

12	**An invalidate command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Invalidate EF_{TNR} 2- Select $DF_{SIMTEST}$\EF_{TNR} 3- Send an update binary command to update the selected file.		1- SW = 9F 13, additional data expected shall be 02 90 00 2- SW = 9F XX 3- SW = 98 10.	
13	**A rehabilitate command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Rehabilitate EF_{TNR} 2- Select $DF_{SIMTEST}$\ EF_{TNR} 3- Send an update binary command to update the selected file.		1- SW = 9F 13, additional data expected shall be 02 90 00 2- SW = 9F XX 3- SW = 90 00.	

5.4.2.1.3 Test coverage

CRR number	Test case number
N1	1, 2,3,4,5,6,7,8,9,10,11,12, 13

5.4.2.2 SIM Output command

Test area reference: RFM_SCC_OUPT

5.4.2.2.1 Conformance requirement

CRRN1: If a command has P3='00', then the UICC shall send back all available response parameter/data.

CRRN2: The standardised following output commands shall be accepted. These commands shall only occur once in a command string and, if present, shall be the last command in the string. The commands are as defined in 3GPP TS 51.011 [3]:

READ BINARY

READ RECORD

GET RESPONSE

CRRN3: The Response Data shall be placed in the Additional Response Data element of the Response Packet.

5.4.2.2.2 Test suites files

Test Script: SIM_RFM_SCC_OUPT_1.scr

SIM_RFM_SCC_OUPT_1.clr

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**A read binary command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}\EF_{TARU}$. - Read Binary command.		1- SW = 9F 14, additional data expected shall be 02 90 00 FF	
2	**A read record command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Read the first record of the selected file in absolute mode		1- SW = 9F 17, additional data expected shall be 03 90 00 55 55 55 55	
3	**A get response command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Increase command with the following data 01 - Get response command with P3=00		1- SW = 9F 19, additional data expected shall be 04 90 00 55 55 56 00 00 01	

5.4.2.2.3 Test coverage

CRR number	Test case number
N1	1,2,3
N2	3

5.4.3 SIM specific behaviour for Response Packets (Using SMS_PP)

Test area reference: RFM_SRP_SMSP

5.4.3.1 Conformance requirements

CRRN1: If PoR is not requested, no data shall be returned by the SIM's RE/RA and the SIM's RE/RA shall indicate to the terminal to issue a RP-ACK.

CRRN2: If PoR is requested, data shall be returned by the SIM. The SIM shall indicate to the terminal to issue a RP-ACK if the response status code octet is '00' or a RP-ERROR if there is a security error of some kind.

CRRN3: The data returned by the SIM is the complete Response Packet to be included in the User Data part of the SMS-DELIVER-REPORT.

CRRN4: If a proof of Receipt is required by the sending entity, the Additional Response Data sent by the Remote File Management Application shall be formatted according to the following table:

Length	Name
1	Number of commands executed within the command script (see note)
2	Last executed command status word
X	Last executed command response data if available (i.e., if the last command was an outgoing command)
NOTE:	This field shall be set to '01' if one command was executed within the command script, '02' if two commands were executed, etc…

5.4.3.2 Test Area Files

Test Script: SIM_RFM_SRP_SMSP_1.scr

SIM_RFM_SRP_SMSP_1.clr

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	APDU Expectation
1	**If PoR is not requested, no data shall be returned** 1- Send a formatted SMS, with no PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select Unknow file. 2- Send a formatted SMS, with no PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with 44 44 44 44 3- Select EF_TARU 4- Read the file content		1- SW = 90 00, no additional data expected 2- SW = 90 00, no additional data expected. 4 SW=90 00, returned data shall be 44 44 44 44	
2	**If PoR is requested, data shall be returned by the SIM.** 1- Send a formatted SMS, with PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{IARU}. - Read record command 2- Perform GetResponse command 3- Send a formatted SMS with a PoR required and a RC/CC/DS error, which contains the following command: - Select $DF_{SIMTEST}$. 4- Perform GetResponse command		1- SW = 9F 17 2- Additional data expected shall be 03 90 00 55 55 55 55. 3- SW = 9E 10 4- Status code expected shall be 01.	

5.4.3.3 Test Coverage

CRR number	Test case number
N1	1
N2	2
N3	2
N4	2

5.5 Remote File Management for USIM

5.5.1 Behaviour of the Remote File Management Application

Test area reference: RFM_URB_CMDS

5.5.1.1 Conformance Requirement

CRRN1: The parameter(s) in the Data Download Message to UICC is either a single command, or a list of commands and shall be processed sequentially.

CRRN2: The application shall take parameters from the Data Download Message to UICC and shall act upon the 3G files according to these parameters.

CRRN3: A Command "session" is defined as starting upon receipt of the parameter/command list, and ends when the parameter list in the Data Download Message to UICC is completed, or when an error is detected which shall halt further processing of the command list.

CRRN4: A command “session” shall not change the logical state (e.g file pointers) of the UICC.

5.5.1.2 Test Area Files

Test Script: USIM_RFM_URB_CMDS_1.scr

USIM_RFM_URB_CMDS_1.clr

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**A command session may contain a single command** 1- Send a formatted SMS which contains a Select $DF_{SIMTEST}$ command on the applet, with a PoR required.			1- SW = 61 13, additional data expected shall be 01 61 XX
2	**A command session may contain multiple commands** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 01 01 01			1- SW = 61 13, additional data expected shall be 03 90 00
3	**A command session end when a error occurs** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select Unknow file. - Select EF_{TARU}. - Update Binary command with data 02 02 02 2- Select EF_TARU 3- Read binary command			1- SW = 61 13, additional data expected shall be 02 6A 82 2- SW = 61 XX 3- SW = 90 00, expected data shall be 01 01 01
4	**A Command session shall not change the logical state of the UICC.** 1- Select $DF_{SIMTEST}$ 2- Select EF_{LARU} 3- Read the selected file using read record command in next access mode. 4- Read the selected file using read record command in next access mode. 5- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Read the selected file using read record command in next access mode with data 6- Read the selected file using read record command in current access mode			3- SW = 90 00, Returned data shall be 55 55 55 554- SW = 90 00, Returned data shall be AA AA AA AA5- SW =61 17, additional data expected shall be 03 90 00 55 55 55 55 6- SW = 90 00, Returned data shall be AA AA AA AA

5.5.1.3 Test Coverage

CRR number	Test case number
N1	1, 2
N2	1,2,3,4
N3	3
N4	4

5.5.2 Coding of the command

5.5.2.1 USIM Input command

Test area reference: RFM_UCC_INPT

5.5.2.1.1 Conformance requirements:

CRRN1: The standardised following input commands shall be accepted. The commands are as defined in 3GPP TS 31.101:

SELECT (SELECT command shall not include the selection by DF name corresponding to P1='04' in the Command)

UPDATE BINARY

UPDATE RECORD

SEARCH RECORD

INCREASE

VERIFY_PIN

CHANGE_PIN

DISABLE_PIN

ENABLE PIN

UNBLOCK PIN

DEACTIVATE FILE

ACTIVATE FILE

5.5.2.1.2 Test suites files

Test Script: USIM_RFM_UCC_INPT_1.scr

USIM_RFM_UCC_INPT_1.clr

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**A select command shall be accepted** 1- Send a formatted SMS which contains a Select $DF_{SIMTEST}$ command with no data return, with a PoR required.			1- SW = 61 13, additional data expected shall be 01 90 00
2	**A select "by path" command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ / EF_{TARU}.			1- SW = 61 13, additional data expected shall be 01 61 XX
3	**An update binary command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{TARU} 3- Update the selected binary command with the following data 00 00 00 4- Send a formatted SMS, with a PoR required in case of error, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Update Binary command with data 01 01 01 5- Select EF_TARU 6- Read binary command			3- SW = 90 00 4- SW = 90 00 3- SW = 90 00, expected data shall be 01 01 01
4	**An update record command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{LARU} 3- Update the selected file with data 00 00 00 00 using update record command in first access mode. 4- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Update the selected file update record command in first access mode with data 01 01 01 01 5- Read the selected file using read record command in current access mode			3- SW = 90 00 4- SW =61 13, additional data expected shall be 03 90 00 5- SW = 90 00, Returned data shall be 01 01 01 01
5	**A search record command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Search record command first-forward access mode with data 01 01 01 01			1- SW =61 13, additional data expected shall be 03 61 01
6	**An increase command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{LARU} 3- Send a formatted SMS, no PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{CARU} - Increase command with the			3- SW = 90 00

	following data 01 4- Read the current selected file using read record command in first access mode.			4- SW = 90 00, returned data shall be 56 56 56.
7	**A verify PIN command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Verify PIN1 with the correct value of the code			1- SW =61 13, additional data expected shall be 03 90 00
8	**A change PIN command shall be accepted** 1- Send a formatted SMS which contains the following commands: - Change PIN1 with value 32 32 32 32 2- Send a verify PIN1 command with the new PIN1 value.			1- SW =90 00 2- SW =90 00
9	**A disable PIN command shall be accepted** 1- Reset the Card. 2- Select EF_{TIAC} (Invalidate access condition set to PIN1). 3- Send an deactivate command 4- Send a formatted SMS with PoR required which contains the following command: - disable PIN1 command 5- Reset the card 6- Select $DF_{SIMTEST}$ 7- Select EF_{TIAC} 8- Send an deactivate command. **9- Send a activate EFTIAC command to reset the file context**			2- SW= 61 XX 3- SW =69 82 4- SW= 61 13, additional data expected shall be 01 90 00 6- SW= 61 XX 7- SW = 61 XX 8- SW = 90 00 9- SW = 90 00
10	**A enable PIN command shall be accepted** 1- Send a formatted SMS with PoR required which contains the following commands: - enable PIN1 command 2- Reset the card 3- Select $DF_{SIMTEST}$\Select EF_{TIAC} 4- Send an deactivate command.			1- SW= 61 13, additional data expected shall be 01 90 00 3- SW = 61 XX 4- SW= 69 82
11	**A unblock PIN command shall be accepted** 1- Block the PIN1 secret code by sending three unsuccessful Verify PIN1 commands. 2- Send a verify PIN1 command with the correct code value. 3- Send a formatted SMS which contains the following commands: - unblock PIN1command 4- Send a verify PIN1 command with the correct code value.			2- SW= 69 83 4- SW= 90 00
12	**A deactivate file command shall be accepted**			

	1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Deactivate EF_{TNR} 2- Select $DF_{SIMTEST}$ 3- Select EF_{TNR} 4- Send an update binary command to update the selected file.			1- SW =61 13, additional data expected shall be 03 90 00 4- SW = 62 83, Selected file invalidate
13	**An activate command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Activate EF_{TNR} 2- Select $DF_{SIMTEST}$ 3- Select EF_{TNR} 4- Send an update binary command to update the selected file.			1- SW =61 13, additional data expected shall be 03 90 00 4- SW = 90 00.

5.5.2.1.3 Test coverage

CRR number	Test case number
N1	1, 2,3,4,5,6,7,8,9,10,11,12, 13

5.5.2.2 USIM Output command

Test area reference: RFM_UCC_OUPT

5.5.2.2.1 Conformance requirements:

CRRN1: If a command has P3='00', then the UICC shall send back all available response parameter/data.

CRRN2: The standardised following output commands shall be accepted. These commands shall only occur once in a command string and, if present, shall be the last command in the string. The commands are as defined in 3GPP TS 31.101:

READ BINARY

READ RECORD

GET RESPONSE

CRRN3: The Response Data shall be placed in the Additional Response Data element of the Response Packet.

5.5.2.2.2 Test Area Files

Test Script: USIM_RFM_UCC_OUPT _1.scr

USIM_RFM_UCC_OUPT _1.clr

Test procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**A read binary command shall be accepted** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$\EF_{TARU}. - Read Binary command.			1- SW = 61 14, additional data expected shall be 02 90 00 FF
2	**A read record command shall be accepted** - Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{LARU} - Read the first record of the selected file in absolute mode			3- SW = 90 00 4- SW = 61 17, additional data expected shall be 03 90 00 55 55 55 55
3	**A get response command shall be accepted** 1- Select $DF_{SIMTEST}$ 2- Select EF_{LARU} 3- Send a formatted SMS, with a PoR required, which contains the following commands: - Select $DF_{SIMTEST}$ - Select EF_{CARU} - Increase command with the following data 01 - Get response command with P3=00			3- SW = 61 17, additional data expected shall be 04 90 00 55 55 56 00 00 01

5.5.2.2.3 Test coverage

CRR number	Test case number
N1	1,2,3
N2	3

5.5.3 USIM specific behaviour for Response Packets (Using SMS_PP)

Test area reference: RFM_URP_SMSP

5.5.3.1 Conformance requirements:

CRRN1: If PoR is not requested, .no data shall be returned by the USIM's RE/RA and the USIM's RE/RA shall indicate to the terminal to issue a RP-ACK. (Note: need for clarification from the 23.048 [6] WG)

CRRN2: If PoR is requested, data shall be returned by the USIM. The USIM shall indicate to the terminal to issue a RP-ACK if the response status code octet is '00' or an RP-ERROR if there is a security error of some kind.

CRRN3: The data returned by the USIM is the complete Response Packet to be included in the User Data part of the SMS-DELIVER-REPORT.

CRRN4: If a proof of Receipt is required by the sending entity, the Additional Response Data sent by the Remote File Management Application shall be formatted according to the following table:

Length	Name
1	Number of commands executed within the command script (see note)
2	Last executed command status word
X	Last executed command response data if available (i.e., if the last command was an outgoing command)
NOTE:	This field shall be set to '01' if one command was executed within the command script, '02' if two commands were executed, etc…

5.5.3.2 Test Area Files

Test Script: USIM_RFM_URP_SMSP_1.scr

USIM_RFM_URP_SMSP_1.clr

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**If PoR is not requested, no data shall be returned** 1- Send a formatted SMS, with no PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select Unknown file. 2- Send a formatted SMS, with no PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{TARU}. - Read Binary command			1- SW = 90 00, no additional data expected 2- SW = 90 00, no additional data expected.
2	**If PoR is requested, data shall be returned by the SIM.** 1- Send a formatted SMS, with PoR required, which contains the following commands: - Select $DF_{SIMTEST}$. - Select EF_{IARU}. - Read record command 2- Perform GetResponse command 3- Send a formatted SMS with a PoR required and a RC/CC/DS error, which contains the following command: - Select $DF_{SIMTEST}$. 4- Perform GetResponse command			1- SW = 61 17 2- Additional data expected shall be 03 90 00 55 55 55 55. 3- SW = 62 00 4- Status code expected shall be 01.

5.5.3.3 Test Coverage

CRR number	Test case number
N1	1
N2	2
N3	2
N4	2

5.6 Remote Applet Management

5.6.1 Remote Applet Management Application behaviour

5.6.1.1 Command session description

Test Area Reference: RAM_RAB_CMDS

5.6.1.1.1 Conformance Requirements

Normal execution

CRRN1: The parameter(s) in the Data Downlaod Message to UICC is either a single command, or a list of commands, which shall be processed sequentially

CRRN2: A Command "session" starts upon the reception of the command list

CRRN3: A Command "session" ends when the parameter list in the Data Download Message to UICC is completed, or when an error is detected which shall halt further processing of the command list

Error case

CRRE1: A Command "session" ends when an error is detected which shall halt further processing of the command list

5.6.1.1.2 Test Area Files

Test Script: (U)SIM_RAM_RAB_CMDS_1.scr

Test Applet: RAM_RAB_CMDS_1.java

Load Script: (U)SIM_RAM_RAB_CMDS_1.ldr

Cleanup Script: (U)SIM_RAM_RAB_CMDS_1.clr

Parameter File: RAM_RAB_CMDS_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	Install of the applet using the load() and install() methods			
1	**A command session may contain a single command** 1- Select the applet 2- Reset the card 3- Send a formatted SMS which contains a Delete command on the applet, with a PoR required on error 4- Select the applet		 1- SW = 9000 3- SW = 9000 4- SW = 6X XX	 1- SW = 9000 3- SW = 9000 4- SW = 6X XX
2	**A command session may contain multiple commands** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - Install and make selectable the applet. - SetSatus to lock the applet.		 1- SW = 9F 13, additional data expected shall be 02 90 00	 1- SW = 61 13, additional data expected shall be 02 90 00
3	**A command session end when an error occurs** 1- Send a formatted SMS, with a PoR required, which contains the following commands: - SetSatus to make selectable the applet. - SetSatus with bad P1. - Delete the applet 2- Select the applet		 1- SW = 9F 13, additional data expected shall be 02 6A 86 2- SW = 90 00	 1- SW = 61 13, additional data expected shall be 02 6A 86 2- SW = 90 00

Note: for the above test cases, the PoR is sent using SMS Deliver-Report.

5.6.1.1.3 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	3
N3	3
E1	3

5.6.1.2 Applet management behaviour

Test Area Reference: RAM_RAB_MNGT

5.6.1.2.1 Conformance Requirements

Normal execution

CRRN1: A loading session consists of a sequence of 1 INSTALL(load) command, 1 or more LOAD(multiple intermediate) commands and 1 LOAD(final) command

CRRN2: Applet installation may only be performed if the corresponding package has already been loaded onto the card

CRRN3: Applet Installation is performed using an INSTALL(install) command

CRRN4: The Package Removal process is performed using a DELETE command

CRRN5: The Applet Removal process is performed using a DELETE command. The UICC shall remove the components that make up the applet

CRRN6: The Applet locking (and unlocking) procedure allows the Network Operator or Service Provider to disable (and enable) an applet using a SET STATUS command

CRRN7: A locked applet could not be triggered or selected

CRRN8: All menu entries of a locked applet will be disabled (i.e. removed from the SET UP MENU command)

CRRN9: An applet Parameters Retrieval procedure allows the Network Operator or Service Provider to remotely request the parameters of an applet, using a GET DATA command

Error case

CRRE1: The card shall reject a package removal command with the corresponding status error code if non removed applications installed from this package remains

CRRE2: The card shall reject a package removal command with the corresponding status error code if the package is referred by other package(s)

5.6.1.2.2 Test Area Files

Test Script: (U)SIM_RAM_RAB_MNGT_1.scr

Test Applet: RAM_RAB_MNGT1_1.java

RAM_RAB_MNGT2_1.java

RAM_RAB_MNGT3_1.java

Load Script: none

Cleanup Script: (U)SIM_RAM_RAB_MNGT_1.clr

Parameter File: RAM_RAB_MNGT1_1.par

RAM_RAB_MNGT2_1.par

RAM_RAB_MNGT3_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**Bad package loading sequence 1** 1- Send SMS to load Package1 using the following sequence: 1 or more LOAD(multiple intermediate) commands 1 LOAD(final) command 2- Send SMS to install Applet1 using INSTALL(install+selectable) 3- Select Applet1		1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX	1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX
2	**Bad package loading sequence 2** 1- Send SMS to load Package1 using the following sequence: 1 INSTALL(load) command 1 LOAD(final) command 2- Send SMS to install Applet1 using INSTALL(install+selectable) 3- Select Applet1		1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX	1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX
3	**Bad package loading sequence 3** 1- Send SMS to load Package1 using the following sequence: 1 INSTALL(load) command 1 or more LOAD(multiple intermediate) commands with 1 LOAD missing 1 LOAD(final) command 2- Send SMS to install Applet1 using INSTALL(install+selectable) 3- Select Applet1		1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX	1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX
4	**Bad package loading sequence 4** 1- Send SMS to load Package1 using the following sequence: 1 INSTALL(load) command 1 or more LOAD(multiple intermediate) commands 2- Send SMS to install Applet1 using INSTALL(install+selectable) 3- Select Applet1		1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX	1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX
5	**Bad package loading sequence 5** 1- Send SMS to load Package1 using the following sequence: 1 LOAD(final) command 1 or more LOAD(multiple intermediate) commands 1 INSTALL(load) command 2- Send SMS to install Applet1 using INSTALL(install+selectable) 3- Select Applet1		1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX	1- SW = 90 00 for each command 2- SW = 90 00 3- SW = 6X XX

6	**Good packages loading** **Package2 imports Package3** **Applets installation** 1- Send SMS to load Package1 using the correct sequence: 1 INSTALL(load) command 1 or more LOAD(multiple intermediate) commands 1 LOAD(final) command 1- Send SMS to load Package3 using the correct sequence: 1 INSTALL(load) command 1 or more LOAD(multiple intermediate) commands 1 LOAD(final) command 3- Send SMS to load Package2 using the correct sequence: 1 INSTALL(load) command 1 or more LOAD(multiple intermediate) commands 1 LOAD(final) command 4- Send SMS to install Applet1 with 1 menu ("Menu1") using INSTALL(install+selectable) 5- Select Applet1 6- Reset the card 7- Send SMS to install Applet2 with 1 menu ("Menu2") using INSTALL(install+selectable) 8- Select Applet2		1- SW = 90 00 for each command 2- SW = 90 00 for each command 3- SW = 90 00 for each command 4- SW = 90 00 5- SW = 90 00 6- SW = 90 00 7- SW = 90 00 8- SW = 90 00	1- SW = 90 00 for each command 2- SW = 90 00 for each command 3- SW = 90 00 for each command 4- SW = 90 00 5- SW = 90 00 6- SW = 90 00 7- SW = 90 00 8- SW = 90 00
7	**Applet locking** 1- Reset the card and perform a Terminal Profile with the menus and Display text facilities 2- Send a SMS to lock Applet1 using SetStatus() 3- Send a Formatted SMS to trigger Applet1 4- Select Applet1	3- Applet1 is not triggered	1- Set Up Menu proactive command is fetched with the menus "Menu1" and "Menu2" 2- SW = 91 2B, Set Up Menu proactive command is fetched with only menu "Menu2" 3- SW = 90 00 4- SW = 6X XX	1- Set Up Menu proactive command is fetched with the menus "Menu1" and "Menu2" 2- SW = 91 2B, Set Up Menu proactive command is fetched with only menu "Menu2" 3- SW = 90 00 4- SW = 6X XX
8	**Applet unlocking** 1- Reset the card and perform a Terminal Profile without the menus and Display text facilities 2- Send a SMS to make selectable Applet1 using SetStatus() 3- Select Applet1 4- Reset the card and perform a Terminal Profile with the menus and Display text facilities 5- Send a Formatted SMS to trigger Applet1	5- Applet1 is triggered	2- SW = 90 00 3- SW = 90 00 4- SW = 91 2B, Set Up Menu proactive command is fetched with the menus "Menu1" and "Menu2" 5- SW = 91 12 and a DISPLAY test proactive session is processed	2- SW = 90 00 3- SW = 90 00 4- SW = 91 2B, Set Up Menu proactive command is fetched with the menus "Menu1" and "Menu2" 5- SW = 91 12 and a DISPLAY test proactive session is processed
9	**Applet removal** 1- Send a SMS to delete Applet1 using Delete() command.		1- SW = 91 22, Set Up Menu proactive command is fetched with only menu "Menu2"	1- SW = 91 22, Set Up Menu proactive command is fetched with only menu "Menu2"
10	**Bad package removal** 1- Reset the card and perform a Terminal Profile without the menus facilities 2- Send a SMS to delete Package2 using Delete() command with a PoR required. 3- Send a SMS to delete Applet2 using Delete()		2- SW = 9F 13, additional data expected shall be 01 69 85	2- SW = 61 13, additional data expected shall be 01 69 85

	command. 4- Send a SMS to delete Package3 using Delete() command with a PoR required.		3- SW = 90 00 4- SW = 9F 13, additional data expected shall be 01 69 85	3- SW = 90 00 4- SW = 61 13, additional data expected shall be 01 69 85
11	**Good package removal** 1- Send a SMS to delete Package2 using Delete() command 2- Send a SMS to delete Package3 using Delete() command 3- Send a SMS to get the status on Package3 using GetStatus() with a PoR required.		1- SW = 90 00 2- SW = 90 00 3- SW = 9F 13, additional data expected shall be 01 6A 88	1- SW = 90 00 2- SW = 90 00 3- SW = 61 13, additional data expected shall be 01 6A 88

Note: for the above test cases, the PoR is sent using SMS Deliver-Report.

5.6.1.2.3 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5
N2	1, 2, 3, 4, 5, 6
N3	1, 2, 3, 4, 5, 6
N4	10, 11
N5	9, 10
N6	7, 8
N7	7, 8
N8	7, 8
N9	Not testable
E1	10
E2	10

5.6.2 Commands coding

5.6.2.1 Commands coding structure

Test Area Reference: RAM_RCC_STRC

5.6.2.1.1 Conformance Requirements

Normal execution

CRRN1: The messages for the Card Manager shall have a TAR value set to '000000' in hexadecimal

CRRN2: Each command is coded according to the generalised structure defined below; each element other than the Data field is a single octet; see 3GPP TS 51.011 [3].

Class byte (CLA)	Instruction code (INS)	P1	P2	P3	Data

CRRN3: If a command has P3='00', then the UICC shall send back all available response parameters/data (case of getResponse)

5.6.2.1.2 Test Area Files

Test Script: (U)SIM_RAM_RCC_STRC_1.scr

Test Applet: RAM_RCC_STRC_1.java

Load Script: (U)SIM_RAM_RCC_STRC_1.ldr

Cleanup Script: (U)SIM_RAM_RCC_STRC_1.clr

Parameter File: RAM_RCC_STRC_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	Install of the package using the load() and install() methods			
1	**The TAR of the Card Manager shall be '00 00 00'** 1- Send a formatted SMS with the TAR set to '01 01 01', to install the applet, using the Install(install+selectable) command. 2- Select the applet 3- Send a formatted SMS with the TAR set to '00 00 00', to install the applet, using the Install(install+selectable) command. 4- Select the applet		1- SW = 90 00 2- SW = 6X XX 3- SW = 90 00 4- SW = 90 00	1- SW = 90 00 2- SW = 6X XX 3- SW = 90 00 4- SW = 90 00

Note: for the above test cases, the PoR is sent using SMS Deliver-Report.

5.6.2.1.3 Test Coverage

CRR number	Test case number
N1	1
N2	Test in sections input and output command coding
N3	Not applicable for RAM output commands (Get Data and Get Status)

5.6.2.2 Input command coding

Test Area Reference: RAM_RCC_INPT

5.6.2.2.1 Conformance Requirements

Normal execution

CRRN1: The Remote Applet Management Application shall process the following input commands: INSTALL, LOAD, DELETE, SET STATUS and PUT KEY

5.6.2.2.2 Test Area Files

Test Script: (U)SIM_RAM_RCC_INPT_1.scr

Test Applet: RAM_RCC_INPT_1.java

Load Script: (U)SIM_RAM_RCC_INPT_1.ldr

Cleanup Script: (U)SIM_RAM_RCC_INPT_1.clr

Parameter File: RAM_RCC_INPT_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**Package Installation** 1- Send a formatted SMS with a PoR required on error to initialise the package loading using the Install(load) command 2- Send all the formatted SMS with a PoR required on error to load the package loading using the Load() command		1- SW = 90 00 2- SW = 90 00	1- SW = 90 00 2- SW = 90 00
2	**Applet installation** 1- Send a formatted SMS with a PoR required on error to install the applet using the Install(install) command 2- Select Applet 3- Reset the card		1- SW = 90 00 2- SW = 90 00	1- SW = 90 00 2- SW = 90 00
3	**SetStatus() command** 1- Send a formatted SMS with a PoR required on error to lock the applet using the SetSatus() command 2- Select Applet		1- SW = 90 00 2- SW = 6X XX	1- SW = 90 00 2- SW = 6X XX

Note: for the above test cases, the PoR is sent using SMS Deliver-Report.

5.6.2.2.3 Test Coverage

CRR number	Test case number
N1	1 to 3

Note: CRRN1 on Put Key command could not be tested in an interoperable way.

5.6.2.3 Output command coding

Test Area Reference: RAM_RCC_OUPT

5.6.2.3.1 Conformance Requirements

Normal execution

CRRN1: The Remote Applet Management Application shall process the following output commands: GET STATUS and GET DATA

CRRN2: For output commands, response Data shall be placed in the Additional Response Data element of the Response Packet. If SMS is being used, these should result in the generation of a single SM by the UICC.

5.6.2.3.2 Test Area Files

None

5.6.2.3.3 Test Procedure

None.

5.6.2.3.4 Test Coverage

CRR number	Test case number
N1	Not Testable, (inconsistencies in specification on case 4 commands)
N2	Not Testable, (inconsistencies in specification on case 4 commands)

5.6.3 (U)SIM Response Packet

Test Area Reference: RAM_SRP_SMSP

5.6.3.1.1 Conformance Requirements

Normal execution

CRRN1: If PoR is requested, data shall be returned by the SIM. The SIM shall indicate to the terminal to issue a RP-ACK if the response status code octet is '00' or a RP-ERROR if there is a security error of some kind.

CRRN2: If PoR is not requested, no data shall be returned by the SIM's RE/RA and the SIM's RE/RA shall indicate to the terminal to issue a RP-ACK.

CRRN3: The data returned by the SIM is the complete Response Packet to be included in the User Data part of the SMS-DELIVER-REPORT.

CRRN4: If a proof of Receipt is required by the sending entity, the Additional Response Data sent by the Remote File Management Application shall be formatted according to the following table:

Length	Name
1	Number of commands executed within the command script (see note)
2	Last executed command status word
X	Last executed command response data if available (i.e., if the last command was an outgoing command)
NOTE:	This field shall be set to '01' if one command was executed within the command script, '02' if two commands were executed, etc…

Note: This last point is not testable due to an inconsistency in the specification on the case 4 commands.

5.6.3.1.2 Test Area Files

Test Script: (U)SIM_RAM_SRP_SMSP_1.scr

Test Applet: RAM_SRP_SMSP_1.java

Load Script: (U)SIM_RAM_SRP_SMSP_1.ldr

Cleanup Script: (U)SIM_RAM_SRP_SMSP_1.clr

Parameter File: RAM_SRP_SMSP_1.par

Test Procedure

Note: for following test cases, the PoR is sent using SMS Deliver-Report

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	Install of the package and the applet using the load() and install() methods			
1	**PoR Required** 1- Send a formatted SMS with a PoR required to lock the applet using the SetStatus() command 2- Send a formatted SMS with a PoR required to make selectable the applet using the SetStatus() command and with a RC/CC/DS error		1- SW = 9F 13, and additional data expected shall be 01 90 00 2- SW = 9F 10 and status code expected shall be 01	1- SW = 61 13, and additional data expected shall be 01 90 00 2- SW = 62 00 and status code expected shall be 01
2	**PoR not required** 1- Send a formatted SMS with a PoR not required to retrieve the applet's status using the GetStatus() command 2- Send a formatted SMS with a PoR not required to retrieve the applet's status using the GetStatus() command and with a RC/CC/DS error.		1- SW = 90 00 2- SW = 90 00	1- SW = 90 00 2- SW = 90 00

Note: forthe above test cases, the PoR is sent using SMS Deliver-Report

5.6.3.1.4 Test Coverage

CRR number	Test case number
N1	1
N2	2
N3	1
N4	Partially covered in 1

5.7 Annex A commands

5.7.1 Applet Management Commands

5.7.1.1 Commands Description

Test Area Reference: *ANA_RAM_CMDS*

5.7.1.1.1 Conformance Requirements

Normal execution

CRRN1: The minimum security applied to a Secured Packet containing Applet Management Commands shall be integrity using CC or DS.

CRRN2: The references to DAP (Data Authentication Pattern) fields are not applicable for Over The Air Application Management, therefore the corresponding data field length for the file DAP shall be set to 0.

Error cases

CRRE1: If the AID is already present in the registry the card shall reject the applet downloading.

5.7.1.1.2 Test suite files

Test Script: (U)SIM_ANA_RAM_CMDS_1.scr

Test Applet: ANA_RAM_CMDS_1.java

Load Script: None.

Cleanup Script: (U)SIM_ANA_RAM_CMDS_1.clr

Parameter File: ANA_RAM_CMDS_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**An install(load) with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error, which contains an Install(load) for the test applet with no security applied.		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level'	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level'
2	**An install(load) with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error, which contains an Install(load) for the test applet with CC applied.		1- SW 90 00	1- SW 90 00
3	**A load command with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error which contains a Load command for the first data block of the test applet with no security applied.		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level'	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level'
4	**A load command with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error which contains a Load command for the first data block of the test applet with CC applied. 2- If necessary: Send all further formatted SMS' with a PoR required on error and with CC applied to load the remainder of the test applet.		1- SW 90 00 2- SW 90 00 (for each SMS)	1- SW 90 00 2- SW 90 00 (for each SMS)
5	**An install(install) with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error, which contains an Install(install) for the test applet with no security applied.		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level'	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level'
6	**An install(install) with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error, which contains an Install(install) for the test applet with CC applied.		1- SW 90 00	1- SW 90 00
7	**An install(makeSelectable) with no security shall fail** 1- Send a formatted SMS with a PoR required on error, which contains an Install(MakeSelectable) for the test applet with no security applied.		1- SW 9F 10, Response Status Code shall be 0A – 'Insufficient Security Level'	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level'
8	**An install(makeSelectable) with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error, which contains an Install(MakeSelectable) for the test applet with CC applied. 2- Select the applet 3- Reset the card		1- SW 90 00 2- SW 90 00	1- SW 90 00 2- SW 90 00
9	**A Set Status command with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error, which contains a Set Status command with no security applied. The command shall set the test applet instance to the state LOCKED. 2- Select the test applet 3- Reset the card		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level' 2- SW 90 00	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level' 2- SW 90 00

10	**A Set Status command with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error, which contains a Set Status command with CC applied. The command shall set the test applet instance to the state LOCKED. 2- Select the applet instance 3- Reset the card		1- SW 90 00 2- SW 6X XX	1- SW 90 00 2- SW 6X XX
11	**A Get Status command with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error, which contains a Get Status command for the test applet instance with no security applied.		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level'	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level'
12	**A Get Status command with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error, which contains a Get Status command for the test applet instance with CC applied. 2- Send a formatted SMS with a PoR required on error, which contains a Set Status command with CC applied. The command shall set the status of the test applet instance to SELECTABLE.		1- SW 90 00 2- SW 90 00	1- SW 90 00 2- SW 90 00
13	**A Delete command with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error, which contains a Delete command for the test applet instance with no security applied. 2- Select the test applet instance. 3- Reset the card		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level' 2- SW 90 00	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level' 2- SW 90 00
14	**A Delete command with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error which contains a Delete command for the test applet instance with CC applied. 2- Select the test applet instance.		1- SW 90 00 2- SW 6X XX	1- SW 90 00 2- SW 6X XX
15	**A Get Data command with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error, which contains a Get Data command for the Card Resources with no security applied.		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level'	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level'
16	**A Get Data command with CC applied shall be successful** 1- Send a formatted SMS with a PoR required on error, which contains a Get Data command for the Card Resources with CC applied.		1. SW 90 00	1. SW 90 00

17	**A PutKey command with no security applied shall fail** 1- Send a formatted SMS with a PoR required on error which contains a Put Key command to replace the KID of the first key set with "FF FF FF FF FF FF FF FF" with no security applied. 2- Send a formatted SMS with a PoR required which contains a SELECT MF command with CC applied. The CC shall be built using the default KID key "01 23 45 67 89 AB CD EF" and shall be checked with the KID of the first key set.		1- SW 9E 10, Response Status Code shall be 0A – 'Insufficient Security Level' 2- SW 9F 13, additional data expected shall be 01 90 00.	1- SW 62 00, Response Status Code shall be 0A – 'Insufficient Security Level' 2- SW 61 13, additional data expected shall be 01 90 00.

Note: for the above tests cases the PoR is sent using SMS Deliver-Report.

7.1.1.1.3 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
E1	Not testable

RAM Command	with CC (successful)	no CC (unsuccessful)
DELETE	14	13
GET DATA	16	15
GET STATUS	12	11
INSTALL(LOAD)	2	1
INSTALL(INSTALL)	6	5
INSTALL(MAKE_SELECTABLE)	8	7
LOAD	4	3
SET STATUS	10	9
PUT KEY	not testable	17

5.7.2 Install commands

5.7.2.1 Install(Load) Command

Test Area Reference: *ANA_INS_LOAD*

5.7.2.1.1 Conformance Requirements

Normal execution

CRRN1: The Load Parameter Field contains a System Parameters Constructed Field TLV (Tag='EF').

CRRN2: The System Parameters Constructed Field TLV is a TLV list compose of:
- A mandatory non volatile memory space required for package loading TLV(Tag='C6',Length='02'),
- an optional non volatile memory requirements for installation TLV (Tag='C8',Length='02')
- an optional volatile memory requirements for installation TLV(Tag='C7',Length='02').

Error cases

CRRE1: If the "Non Volatile memory space for package loading" is not available on the card, the applet loading must be rejected with the Status Word '6A 84', not enough memory space.

CRRE2: If the "Non Volatile memory required for installation" is not available on the card, the applet loading must be rejected with the Status Word '6A 84', not enough memory space.

CRRE3: If the "Volatile memory required for installation" is not available on the card, the applet loading must be rejected with the Status Word '6A 84', not enough memory space.

CRRE4: If the System Parameters Constructed Field TLV list is incorrect (incorrect length or mandatory field missing), the applet loading must be rejected (with the Status Word '6A80', incorrect parameters in data field or '6700', incorrect length).

5.7.2.1.2 Test Area Files

Test Script: (U)SIM_ANA_INS_LOAD_1.scr

Test Applet: ANA_INS_LOAD_1.java

Load Script: none

Cleanup Script: none

Parameter File: ANA_INS_LOAD_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
1	**Install(Load) with good System Parameters (tag C6 and tag C8 included)** 1-Send an Install (load) with good System Parameters(with mandatory tag C6 and optional tag C8 included): EF08C60200FFC8020000 2-Send formatted SMS to load and Install the applet. 3-Select the applet. 4- Reset the card 5- Delete the applet instance and the package.		1- SW = 90 00 2- SW = 90 00 3- SW = 90 00 5- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- SW = 90 00 5- SW = 90 00
2	**Install(Load) with good System Parameters (tag C6, tag C8 and tag C7 included)** 1- Send a formatted SMS, which contains an Install (load) with good System Parameters(with mandatory tag C6, optional tag C8 and optional tag C7 included): EF0CC60200FFC8020000C7020090 2-Send formatted SMS to load and Install the applet. 3-Select the applet. 4- Reset the card 5- Delete the applet instance and the package.		1- SW = 90 00 2- SW = 90 00 3- SW = 90 00 5- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- SW = 90 00 5- SW = 90 00
3	**Install(Load) with incorrect System Parameters (C6 Tag length incorrect)** 1- Send a formatted SMS, with a PoR required, which contains an Install (load) with incorrect System Parameters (C6 Tag length incorrect): EF0DC6030000FFC8020000C7020090 2- Reset the card		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
4	**Install(Load) with incorrect System Parameters (C8 Tag length incorrect)** 1- Send a formatted SMS, with a PoR required, which contains an Install (load) with incorrect System Parameters(C8 Tag length incorrect): EF0EC60200FFC80400000000C7020090 2- Reset the card		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
5	**Install(Load) with incorrect System Parameters (C7 Tag length incorrect)** 1- Send a formatted SMS, with a PoR required, which contains an Install (load) with incorrect System Parameters (C7 Tag length incorrect): EF0EC60200FFC8020000C70400000090 2- Reset the card		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80

Note: for the above tests cases the PoR is send using SMS Deliver-Report.

5.7.2.1.3 Test Coverage

CRR number	Test case number
N1	1,2
N2	1,2
E1	Not testable
E2	Not testable
E3	Not testable
E4	3,4,5

5.7.2.2 Install (install) and install(install and make selectable) commands

Test Area Reference: *ANA_INS_INMS*.

5.7.2.2.1 Conformance Requirements

Normal execution

CRRN1: If the install(install) command is used, the registration is not active (the applet cannot be selected, triggered and its menu entries are not visible).

CRRN2: If the install(install and make selectable) command is used, the registration is active(the applet can be selected, triggered and its menu entries are visible).

CRRN3: The applet shall be registered with the instance AID present in the Install(install) command.

CRRN4: The Install Parameters Field is a TLV list composed of a System Parameters TLV (Tag = 0xEF) and an Applet Specific Parameters TLV (Tag = 0xC9)

CRRN5: For a toolkit application installation, the System Parameters TLV is composed of:
- a Non Volatile Memory Requirements for Installation TLV (Tag = 0xC8, length = 0x02)
- a Volatile Memory Requirements for Installation TLV (Tag = 0xC7, length = 0x02)
- a Toolkit Applet Specific Parameters TLV (Tag = 0xCA)

CRRN6: If an optional parameter of Applet Specific Parameters is included, then all the parameters shall be included also

CRRN7: For the applet installation, the card shall used the values defined in the Install(install) command for the Non Volatile Memory Requirement and the Volatile Memory Requirement and not the one of the Install(load) command.

CRRN8: The Toolkit Applet Specific Parameters is composed of mandatory and optional field:
- the Access Domain field (mandatory),
- the priority level of the toolkit applet instance field (mandatory),
- the maximum number of timers allowed for the applet instance (mandatory),
- the maximum text length for a menu entry (mandatory),
- the maximum number of menu entries allowed for the applet instance (mandatory),
- the position and the identifier of each menu entry (optional, must be present if the maximum number of menu entry is not null),
- the maximum number of channels for the applet instance (optional, must be present if the minimum security level field is present).
- the minimum security level field (optional, must be present if the maximun number of channels field is present).

CRRN9: the Access Domain field is composed of:
- the length of the field (mandatory, minimum is 1)
- the Access Domain Parameter (mandatory),
- the Access Domain Data (optional).

CRRN10: The Access Domain parameter indicates the mechanism used to control the applet instance access to the GSM file System (‘00’ means full access to the GSM File System, ‘FF’ means no access to the GSM File System).

CRRN11: The priority specifies the order of activation of an applet compared to the other applet registered to the same event ('01': Highest priority level, 'FF': Lowest priority level)

CRRN12: If two or more applets are registered to the same event and have the same priority level, the applets are activated according to their installation date (i.e. the most recent applet is activated first)

CRRN13: the maximum number of timers allowed for the applet instance is defined at the installation of the toolkit applet.

CRRN14: The maximum length of item text string is defined at the installation of the toolkit applet.

CRRN15: The maximum number of menu entries is defined at the installation of the toolkit applet and each correct call to initMenuEntry() up to this maximum shall be successful.

CRRN16: The position of the new menu entries is an absolute position among the existing ones.

CRRN17: If the position identifier is 00h, the menu shall have the last position

CRRN18: If the requested item identifier in the range [1-127] is not already allocated, then this item identifier shall be allocated to the current applet.

CRRN19: If the requested item identifier is '00', then the card shall take the first free value in the range [128,255].

CRRN20: The Receiving Entity shall check the Minimum Security Level during processing the security of the Command Packet.

CRRN21: The Receiving Entity shall reject the message if the MSL check fails, and a Response Packet with the 'Insufficient Security Level' Response Status Code shall be sent when required.

CRRN22: If the length of the Minimum Security Level field is greater than zero, the Minimum Security Level is used to specify the minimum level of security to be applied to Secured Packets. The first byte shall be the MSL Parameter, other bytes shall be the MSL Data

CRRN23: If the length of the Minimum Security Level field is zero, no minimum security level check shall be performed by the receiving entity.

CRRN24: If no Minimum Security Level field is present (no MSL length, no MSL parameter and no MSL data), no minimum security level check shall be performed by the receiving entity.

CRRN25: If the Maximum number of channels field is included in the command data then the Length of Minimum Security Level field shall also be included.

Error cases

CRRE1: For a toolkit applet installation, if the Install Parameters Field TLV list or the System Parameters TLV list are incorrect (incorrect length or mandatory field missing), the installation failed (with the Status Word '6A80', incorrect parameters in data field or ‘6700’, incorrect length) .

CRRP2: If the Access Domain Parameter requested is not supported, the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

CRRE3: If an applet with Access Domain Parameter 'FF' (i.e. No Access to the GSM File System) tries to access a GSM file (e.g. invoke the updateBinary(..) method) the framework shall throw a SIMViewException with a AC_NOT_FULFILLED reason.

CRRE4: If an applet with Access Domain Parameter '00' (i.e. Full Access to the GSM File System) tries to access a GSM file (e.g. invoke the updateBinary(..) method) with an access condition set to NEVER, the framework shall throw a SIMViewException with a AC_NOT_FULFILLED reason.

CRRE5: The total number of timers allocated for all the applets shall not exceed 8. If the maximum number of timers required is greater than '08' (maximum numbers of timers specified in 3GPP TS 31.111 [4], the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

CRRE6: If the requested item identifier is in the range [128,255], then the card shall reject the install command.

CRRE7: The total number of channels allocated for all the applets shall not exceed 7. If the maximum number of channels required is greater than '07' (maximum numbers of channels specified in 3GPP TS 31.111 [4]), the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

CRRE8: If the AID present in the Install(install) command and in the Load() command are different, the installation fails.

CRRE9: If the register (bArray, bOffset, bLength) is invoked with an AID passed in the parameters different from the instance AID provided in the install method buffer, the registration fails.

CRRE10: If the memory (volatile or non volatile) requested in the Install(install) command is insufficient, the installation fails.

CRRE11: If the requested item identifier in the range [1-127] is already allocated, then the card shall reject the install command.

5.7.2.2.2 Test Area Files

Test Script: (U)SIM_ANA_INS_INMS_1.scr

Test Applet: ANA_INS_INMS_1.java

ANA_INS_INMS_2.java

ANA_INS_INMS_3.java

ANA_INS_INMS_4.java

ANA_INS_INMS_5.java

ANA_INS_INMS_6.java

ANA_INS_INMS_7.java

ANA_INS_INMS_8.java

ANA_INS_INMS_9.java

ANA_INS_INMS_10.java

ANA_INS_INMS_11.java

Load Script: (U)SIM_ANA_INS_INMS_1.ldr

Cleanup Script: (U)SIM_ANA_INS_INMS_1.clr

Parameter File: ANA_INS_INMS_1.par

Test Procedure

Id	Description	API Expectation	SIM APDU Expectation	USIM APDU Expectation
0	Load of the Applet1, Applet2, Applet3, Applet4, Applet5, Applet6, Applet7, Applet8 and Applet9 using the load() method			
1	**Install(Install) command** 1- Reset the Card and send terminal profile, with the menu and display text facilities 2-Send a formatted SMS, which contains an Install(Install) of the Applet1 3- Send a formatted SMS to trigger the applet(when the applet is triggered it sends a display text command) 4- Select Applet1 5- Reset the Card 6-Send terminal profile, without the menu facilities 7- Send a formatted SMS to delete Applet1 instance using Delete() command.		 1- SW = 90 00 2- SW = either 90 00 or 91 1B (empty set up menu is processed) 3- SW = 90 00 4- SW = 6X XX 6- SW = 90 00 7- SW = 90 00	 1- SW = 90 00 2- SW = either 90 00 or 91 1B (empty set up menu is processed) 3- SW = 90 00 4- SW = 6X XX 6- SW = 90 00 7- SW = 90 00
2	**Install(Install and Make Selectable) command** 1- Reset the Card and send terminal profile, with the menu and display text facilities 2-Send a formatted SMS, which contains an Install(Install and make selectable) of the Applet1 Load File AID: A0000000090002FFFFFFFF8902000000 AID witin Load File: A0000000090002FFFFFFFF8902001001 Application Instace Identifier: A0000000090002FFFFFFFF8902001102 3- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01' and '02' and send terminal response 4- Send a formatted SMS to trigger the applet 5-Fetch a display text proactive command and send terminal response 6- Select Applet1 7- Reset the Card 8-Send terminal profile, without the menu facilities 9- Send a formatted SMS to delete Applet1 instance using Delete() command.		 1- SW = 90 00 2- SW = 91 29 3- SW = 90 00 4- SW = 91 14 5- SW = 90 00 6- SW = 90 00 8- SW = 90 00 9- SW = 90 00	 1- SW = 90 00 2- SW = 91 29 3- SW = 90 00 4- SW = 91 14 5- SW = 90 00 6- SW = 90 00 8- SW = 90 00 9- SW = 90 00
3	**Applet Installation with Full Install Parameters Field without optional parameters** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet2 with a Full Install Parameters Field without optional parameters:		 1- SW = 90 00	 1- SW = 90 00

	EF14C8020800C7020010CA0A010001001002010 10202C900 2- Select Applet2 3-Reset the Card. 4- Send a formatted SMS to delete Applet2 instance using Delete() command		2- SW = 90 00 4- SW = 90 00	2- SW = 90 00 4- SW = 90 00
4	**Applet Installation with Full Install Parameters Field with all the optional parameters** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet2 with a Full Install Parameters Field with all the optional parameters: EF16C8020800C7020010CA0C010001000002010 102020200C900 2- Select Applet2 3-Reset the Card. 4- Send a formatted SMS to delete Applet2 instance using Delete() command		1- SW = 90 00 2- SW = 90 00 4- SW = 90 00	1- SW = 90 00 2- SW = 90 00 4- SW = 90 00
5	**Applet Installation with access Domain Parameter equal to 00** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet3 with Access Domain Parameter equal to 00 2- Send an Unformatted SMS 3-Read EF_{TARU} 4- Send a formatted SMS to delete Applet3 instance using Delete() command.	2- Applet3 is triggered and calls sim.access.SIMView .select method EF_{TARU} and sim.access.SIMView .updateBinary method with 010101. No exception is thrown	1- SW = 90 00 2- SW = 90 00 3- Value read from EF_{TARU} = 010101 4- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- Value read from EF_{TARU} = 010101 4- SW = 90 00
6	**Applet Installation with access Domain Parameter equal to FF** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet3 with Access Domain Parameter equal to FF 2- Send an Unformatted SMS 3-Read EF_{TARU} 4- Send a formatted SMS to delete Applet3 instance using Delete() command.	2- Applet3 is triggered and calls sim.access.SIMView .select method EF_{TARU} and sim.access.SIMView .updateBinary method with 020202. The sim.access.SIMView Exception AC_NOT_FULFILLED is thrown	1- SW = 90 00 2- SW = 90 00 3- Value read from EF_{TARU} = 010101 4- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- Value read from EF_{TARU} = 010101 4- SW = 90 00
7	**The order of activation of the applets is**			

	specified by the Priority Level 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet4 with Priority Level 01 2-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet 5 with Priority Level FF 3- Send a Unformatted SMS 4- Fetch the display text proactive command of Applet 4 and send terminal response 5- Fetch the display text proactive command of Applet 5 and send terminal response 6- Send a formatted SMS to delete Applet4 instance using Delete() command. 7- Send a formatted SMS to delete Applet5 instance using Delete() command.	3- Applet4 is triggered and then Applet5 is triggered	1- SW = 90 00 2- SW = 90 00 3- SW = 91 14 4- SW = 91 14 5- SW = 90 00 6- SW = 90 00 7- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- SW = 91 14 4- SW = 91 14 5- SW = 90 00 6- SW = 90 00 7- SW = 90 00
8	**The order of activation of the applets is specified by the installation date if the Priority Level is the same** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet4 with Priority Level 01 2-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet5 with Priority Level 01 3- Send a Unformatted SMS 4- Fetch the display text proactive command of Applet 5 and send terminal response 5- Fetch the display text proactive command of Applet 4 and send terminal response 6- Send a formatted SMS to delete Applet4 instance using Delete() command. 7- Send a formatted SMS to delete Applet5 instance using Delete() command.	3- Applet5 is triggered and then Applet4 is triggered	1- SW = 90 00 2- SW = 90 00 3- SW = 91 14 4- SW = 91 14 5- SW = 90 00 6- SW = 90 00 7- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- SW = 91 14 4- SW = 91 14 5- SW = 90 00 6- SW = 90 00 7- SW = 90 00
9	**Maximum number of timer is specified at the installation** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet10 with maximum number of timer 02 2-Trigger Applet10 (formatted SMS) 3-Fetch the display text command containing "TestOK" and send terminal response 3- Send a formatted SMS to delete Applet10 instance using Delete() command.	2-The applet sends a display text "TestOK" if the registration of 2 timers is successful and the registration of a third timer throws an exception	1- SW = 90 00 2- SW = 91 14 3- SW = 90 00 4- SW = 90 00	1- SW = 90 00 2- SW = 91 14 3- SW = 90 00 4- SW = 90 00
10	**Maximum length of item text string is specified**		1- SW = 90 00	1- SW = 90 00

	at the installation 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet11 with maximum length of item text string 10 and maximum number of menu entries 02 2- Trigger Applet11 (formatted SMS) 3-Fetch the display text command containing "TestOK" and send terminal response 4- Send a formatted SMS to delete Applet11 instance using Delete() command.	2-The applet sends a display text "TestOK" if the installation of 1 menu entry with text string length equal to 10 is successful and the installation of the second menu entry with text string length equal to 11 throws an exception	2- SW = 91 14 3- SW = 90 00 4- SW = 90 00	2- SW = 91 14 3- SW = 90 00 4- SW = 90 00
11	**Applet installation with Menu Entries 1** 1-Send terminal profile, with the menu facilities 2-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet6 Install Parameter Field: EF14C8020800C7020010CA0A010001001002010 10202C900 For Applet6 (item id 01): MenuEntry = "Applet 1A" Position = 01 For Applet6 (item id 02): MenuEntry = "Applet 1B" Position = 02 3- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01' and '02' and send terminal response 4-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet7 Install Parameter Field: EF14C8020800C7020010CA0A010001001002030 30404C900 For Applet7 (item id 03): MenuEntry = "Applet 2A" Position = 03 For Applet7 (item id 04): MenuEntry = "Applet 2B" Position = 04 5- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01', '02, '03' and '04' and send terminal response 6- Trigger Applet6 (formatted SMS) 7- Fetch the display text command containing "TestOK" and send terminal response 8- Send a formatted SMS to delete Applet6 instance using Delete() command. 9- Fetch a Set Up Menu proactive command with Menu Entry ID entry '03' and '04' and send terminal	6-The applet sends a display text "TestOK" if the registration of 2 menu entries is successful and the registration of a third menu entry throws an exception	1- SW = 90 00 2- SW = 91 31 3- SW = 90 00 4- SW = 91 49 5- SW = 90 00 6- SW = 91 14 7- SW = 90 00 8- SW = 91 31 9- SW = 90 00 10- SW = 91 1B 11- SW- 90 00	1- SW = 90 00 2- SW = 91 31 3- SW = 90 00 4- SW = 91 49 5- SW = 90 00 6- SW = 91 14 7- SW = 90 00 8- SW = 91 31 9- SW = 90 00 10- SW = 91 1B 11- SW- 90 00

	response 10- Send a formatted SMS to delete Applet7 instance using Delete() command. 11- Fetch an empty Set Up Menu proactive command and send terminal response			
12	**Applet installation with Menu Entries 2** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet6 For Applet6 (item id 01): MenuEntry = ”Applet 1A” Position = 03 For Applet6 (item id 02): MenuEntry = ”Applet 1B” Position = 04 2- Fetch a Set Up Menu proactive command with Menu Entry ID entry ‘01’ and ‘02’ and send terminal response 3-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet7 For Applet7 (item id 03): MenuEntry = ”Applet 2A” Position = 01 For Applet7 (item id 04): MenuEntry = ”Applet 2B” Position = 02 4- Fetch a Set Up Menu proactive command with Menu Entry ID entry ‘03’, ‘04, ‘01’ and ‘02’ and send terminal response 5- Send a formatted SMS to delete Applet6 instance using Delete() command. 6- Fetch a Set Up Menu proactive command with Menu Entry ID entry ‘03’ and ‘04’ and send terminal response 7- Send a formatted SMS to delete Applet7 instance using Delete() command. 8- Fetch an empty Set Up Menu proactive command and send terminal response		1- SW = 91 31 2- SW = 90 00 3- SW = 91 49 4- SW = 90 00 5- SW = 91 31 6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00	1- SW = 91 31 2- SW = 90 00 3- SW = 91 49 4- SW = 90 00 5- SW = 91 31 6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00
13	**Applet installation with Menu Entries 3** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet6 For Applet6 (item id 01): MenuEntry = ”Applet 1A” Position = 01 For Applet6 (item id 02): MenuEntry = ”Applet 1B” Position = 03		1- SW = 91 31	1- SW = 91 31

	2- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01' and '02' and send terminal response		2- SW = 90 00	2- SW = 90 00
	3-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet7 For Applet7 (item id 03): MenuEntry = "Applet 2A" Position = 02 For Applet7 (item id 04): MenuEntry = "Applet 2B" Position = 04		3- SW = 91 49	3- SW = 91 49
	4- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01', '03, '02' and '04' and send terminal response		4- SW = 90 00	4- SW = 90 00
	5- Send a formatted SMS to delete Applet6 instance using Delete() command.		5- SW = 91 31	5- SW = 91 31
	6- Fetch a Set Up Menu proactive command with Menu Entry ID entry '03' and '04' and send terminal response		6- SW = 90 00	6- SW = 90 00
	7- Send a formatted SMS to delete Applet7 instance using Delete() command.		7- SW = 91 1B	7- SW = 91 1B
	8- Fetch an empty Set Up Menu proactive command and send terminal response		8- SW = 90 00	8- SW = 90 00
14	**Applet installation with Menu Entries 4**			
	1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet6 For Applet6 (item id 01): MenuEntry = "Applet 1A" Position = 02 For Applet6 (item id 02): MenuEntry = "Applet 1B" Position = 04		1- SW = 91 31	1- SW = 91 31
	2- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01' and '02 and send terminal response		2- SW = 90 00	2- SW = 90 00
	3-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet7 For Applet7 (item id 03): MenuEntry = "Applet 2A" Position = 01 For Applet7 (item id 04): MenuEntry = "Applet 2B" Position = 03		3- SW = 91 49	3- SW = 91 49
	4- Fetch a Set Up Menu proactive command with Menu Entry ID entry '03', '01, '04' and '02' and send terminal response		4- SW = 90 00	4- SW = 90 00
	5- Send a formatted SMS to delete Applet6		5- SW = 91 31	5- SW = 91 31

	instance using Delete() command. 6- Fetch a Set Up Menu proactive command with Menu Entry ID entry '03' and '04' and send terminal response 7- Send a formatted SMS to delete Applet7 instance using Delete() command. 8- Fetch an empty Set Up Menu proactive command and send terminal response		6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00	6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00
15	**Applet installation with Menu Entries with position identifier equal to 00** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet6 For Applet6 (item id 01): MenuEntry = "Applet 1A" Position = 00 For Applet6 (item id 02): MenuEntry = "Applet 1B" Position = 00 2- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01' and '02' and send terminal response 3-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet7 For Applet7 (item id 03): MenuEntry = "Applet 2A" Position = 00 For Applet7 (item id 04): MenuEntry = "Applet 2B" Position = 00 4- Fetch a Set Up Menu proactive command with Menu Entry ID entry '01', '02, '03' and '04' and send terminal response 5- Send a formatted SMS to delete Applet6 instance using Delete() command. 6- Fetch a Set Up Menu proactive command with Menu Entry ID entry '03' and '04' and send terminal response 7- Send a formatted SMS to delete Applet7 instance using Delete() command. 8- Fetch an empty Set Up Menu proactive command and send terminal response		1- SW = 91 31 2- SW = 90 00 3- SW = 91 49 4- SW = 90 00 5- SW = 91 31 6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00	1- SW = 91 31 2- SW = 90 00 3- SW = 91 49 4- SW = 90 00 5- SW = 91 31 6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00
16	**Applet installation with Menu Entries with item identifier equal to 00** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet6 For Applet6 (item id 00):		1- SW = 91 31	1- SW = 91 31

	MenuEntry = ”Applet 1A” Position = 01 For Applet6 (item id 00): MenuEntry = ”Applet 1B” Position = 02 2- Fetch a Set Up Menu proactive command with Menu Entry ID entry ‘80’(Applet1A), ‘81(Applet 1B) and send terminal response 3-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet7 For Applet7 (item id 00): MenuEntry = ”Applet 2A” Position = 03 For Applet7 (item id 00): MenuEntry = ”Applet 2B” Position = 04 4- Fetch a Set Up Menu proactive command with Menu Entry ID entry ‘80’(Applet1A), ‘81(Applet 1B), ‘82’(Applet 2A) and ‘83’(Applet 2B) and send terminal response 5- Send a formatted SMS to delete Applet6 instance using Delete() command. 6- Fetch a Set Up Menu proactive command with Menu Entry ID entry ‘82’(Applet2A), ‘83(Applet 2B). and send terminal response 7- Send a formatted SMS to delete Applet7 instance using Delete() command. 8- Fetch an empty Set Up Menu proactive command and send terminal response		2- SW = 90 00 3- SW = 91 49 4- SW = 90 00 5- SW = 91 31 6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00	2- SW = 90 00 3- SW = 91 49 4- SW = 90 00 5- SW = 91 31 6- SW = 90 00 7- SW = 91 1B 8- SW = 90 00
17	**Applet installation with MSL length equal to 00** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet8 with maximum number of channels equal to 00 and length of MSL field equal to 00 2- Select Applet8 3- Reset the card 4- Send a Formatted SMS with SPI1 equal to 00 and PoR required 5- Send a formatted SMS to delete Applet8 instance using Delete() command.		1- SW = 90 00 2- SW = 90 00 4- SW = 9F 10, a Response Packet is sent with Status Code equal to 00 5- SW = 90 00	1- SW = 90 00 2- SW = 90 00 4- SW = 62 00, a Response Packet is sent with Status Code equal to 00 5- SW = 90 00
18	**Applet installation with no MSL field** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet8 with no MSL field 2- Select Applet8 3- Reset the card 4- Send a Formatted SMS with SPI1 equal to 00		1- SW = 90 00 2- SW = 90 00 4- SW = 9F 10, a Response	1- SW = 90 00 2- SW = 90 00 4- SW = 62 00, a Response

	and PoR required 5- Send a formatted SMS to delete Applet8 instance using Delete() command.		Packet is sent with Status Code equal to 00 5- SW = 90 00	Packet is sent with Status Code equal to 00 5- SW = 90 00
19	**Applet installation with missing C8 tag** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with missing C8 tag: EF10C7020010CA0A010001001002010102 02C900		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
20	**Applet installation with missing C7 tag** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with missing C7 tag: EF10C8020800CA0A010001001002010102 02C900		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
21	**No access to a GSM file with access conditions set to Never** 1-Send a formatted SMS, which contains an Install(Install and Make Selectable) of the Applet3 with Access Domain Parameter equal to 00 2- Send an Unformatted SMS 3- Read EF_{TNU} 4- Send a formatted SMS to delete Applet3 instance using Delete() command.	2-Applet3 is triggered and calls sim.access.SIMView .select method with EF_{TNU} and sim.access.SIMView .updateBinary method with 030303. The sim.access.SIMView Exception AC_NOT_FULFILLED is thrown	1- SW = 90 00 2- SW = 90 00 3- Value read from EF_{TNU} = 55 55 55 4- SW = 90 00	1- SW = 90 00 2- SW = 90 00 3- Value read from EF_{TNU} = 55 55 55 4- SW = 90 00
22	**Incorrect maximum number of timers** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with 9 numbers of timers: EF10C8020800 C7020010CA0A010001091002010102 02C900		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
23	**Incorrect item identifier** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with incorrect item identifier: EF10C8020800 C7020010CA0A010001001002018*5*0202C900		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
24	**Incorrect numbers of channels** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with incorrect numbers		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80

	of channels: EF12C8020800C7020010CA08010001000000100 0C900			
25	**Differents AIDs in the Install(install) command and in the Load() command** 1-Send a formatted SMS which contains an Install (Install and Make Selectable) of Applet2 with AID present in the Install(install) command and in the Load() command different 2-Select the Applet2		1- SW = 90 00 2- SW = 6X XX	1- SW = 90 00 2- SW = 6X XX
26	**Differents AID in the Install(install) command and in the register** 1-Send a formatted SMS which contains an Install (Install and Make Selectable) of Applet9 with AID present in the Install(install) command and in the register(bArray, bOffset, bLength) different 2-Select the Applet9		1-SW = 90 00 2- SW = 6X XX	1-SW = 90 00 2- SW = 6X XX
27	**Item identifier already allocated** 1-Send a formatted SMS, which contains the following command: -Install (Install and Make Selectable) of the Applet1.In this command the item identifier 2 is reserved. 2-Send a formatted SMS, with a PoR required, which contains the following command: -Install (Install and Make Selectable) of the Applet2.In this command the item identifier 2 is reserved. 3- Send a formatted SMS to delete Applet1 instance using Delete() command.		1- SW = 90 00 2- SW = 9F 13, additional data expected shall be 01 6A 80 3- SW = 90 00	1- SW = 90 00 2- SW = 61 13, additional data expected shall be 01 6A 80 3- SW = 90 00
28	**Applet installation with length incorrect of C8 Tag** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with incorrect length of C8 tag: EF15C803000800C7020010 CA0A010001001002010102 02C900		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80
29	**Applet installation with length incorrect of C7 tag** 1-Send a formatted SMS, with a PoR required, which contains an Install (Install and Make Selectable) of the Applet2 with incorrect length of C7): EF15C8020800C703000010 CA0A010001001002010102 02C900		1- SW = 9F 13, additional data expected shall be 01 6A 80	1- SW = 61 13, additional data expected shall be 01 6A 80

Note: for the above tests cases the PoR is send using SMS Deliver-Report

5.7.2.2.3 Test Coverage

CRR number	Test case number
N1	1
N2	2
N3	2
N4	3, 4
N5	3, 4
N6	4
N7	Not testable
N8	4
N9	5, 6
N10	5, 6
N11	7
N12	8
N13	9
N14	10
N15	11
N16	11, 12, 13, 14
N17	15
N18	11, 12, 13, 14
N19	16
N20	Not Testable
N21	Not Testable
N22	Not Testable
N23	17
N24	18
N25	4
E1	19, 20, 28, 29
E2	Not Testable
E3	6
E4	21
E5	22
E6	23
E7	34
E8	25
E9	26
E10	Not Testable
E11	27

5.7.3 Delete command

No specific functionalities introduced by the TS 3GPP 23.048 [6]

5.7.4 Load command

No specific functionalities introduced by the TS 3GPP 23.048 [6]

5.7.5 Put Key command

5.7.5.1 Command session description

Test Area Reference: ANA_CMD_PUTK

5.7.5.1.1 Conformance Requirements

Normal execution

CRRN1: For a card in post issuance state, the Put Key command is able to update the transport security keys KID and KIc of an existing key set version.

CRRN2: Key indexes 1 and 2 are used for Kic and KID transport security keys and key index 3, KIK, is used to encrypt the key data value.

Error Case

CRRE1: For a card in post issuance state, the Put Key is not able to create a new key set version.

CRRE2: In post issuance state the Put Key command is not able to update the KIK value of an existing key set version.

5.7.5.1.2 Test Area Files

None.

5.7.5.1.3 Test Procedure

None.

5.7.5.1.4 Test Coverage

CRR number	Test case number
N1	Not testable
N2	Not testable
E1	Not testable
E2	Clarification pending

5.7.6 Set Status command

No specific functionalities introduced by the TS 3GPP 23.048 [6]

Annex A (normative): Test area reference acronym table

SEC	Security		
	GSP	Generalised Secured Packet testing	
		CMD	Command Packet structure
		RESP	Response Packet Struture
	SPP	SMS PP Implementation	
		SSS	Structure of SMS PP
		CSS	Command Packet containing a Single SMS
		CCS	Command Packet containing a Concatenated SMS
		RPS	Response Packet Structure
		SMC	Security Mechanism for Command Packet
		SMR	Security Mechanism for Reponse Packet
	SCB	SMS CB Implementation	
		SCB	Structure of SMS CB
		CCB	Command Packet containing in a SMS CB
		SMC	Security Mechanism for Command Packet
RFM	Remote File Management		
	SRB	SIM Remote file management Behaviour	
		CMDS	Command session description
	SCC	SIM Command Coding	
		INPT	Input Commands
		OUPT	Output commands
	SRP	SIM Response Packet	
		SMSP	SMS PP Response Packet
	URB	USIM R Remote file management Behaviour	
		CMDS	Command session description
	UCC	USIM Command Coding	
		INPT	Input Commands
		OUPT	Output commands
	URP	USIM Response Packet	
		SMSP	SMS PP Response Packet
RAM	Remote Applet Management		

	RAB	Remote Applet management Behaviour	
		CMDS	Command session description
		MNGT	Applet Management Behaviour
	RCC	Command Coding	
		STRC	Command Coding Structure
		INPT	Input Command Coding
		OUPT	Output Command Coding
	SRP	(U)SIM Response Packet	
		SMSP	SMS PP Response Packet
ANA	Annex A		
	INS	Install commands	
		LOAD	Install for Load Command
		INMS	Install for Instal and Make Selectable Command
	CMD	Commands	
		PUTK	Put Key Command
		GETD	Get Data Command
	RAM	Remote Applet Management Commands	
		CMDS	Command session description

Annex B (normative): Script file syntax and format description

The syntax used for this test suite is based on the one defined in the 3GPP TS 51.013 Rel-5 [9].

B.1 Syntax description

Following is a syntax description in BNF.

<statement list> ::= [<statement> **\n**] +

<statement> ::= <simple> | <switch> | <*blank line*>

<simple> ::= <Mode> <reset> | <init> | <command> | <remark>

<Mode> ::= **SIM** | **USIM**

<reset> ::= **RST**

<init> ::= **INI** | <*data*>

<command> ::= **CMD** <*data*> [<response>] (<status>)

<response> ::= **[** <*data*> **]**

<status> ::= **(** <*hexbyte*><*hexbyte*> **)**

<data> ::= [< hexbyte >]+

<remark> ::= **REM** <*text line*>

<switch> ::= **SWI { [**<labelled list>] + **}**

<labelled list> ::= <label> **: \n** <statement list>

Description of syntax metalanguage :

\n	represents a linebreak
[x]	means x can appear optionally
[x] +	means 1 or more appearances of x
x | y	means x or y
[ ] { } :	(bold) these are characters that appear literally in the script files
<*text line*>	any character until the end of the line
<*blank line*>	a line containing no text is acceptable
<*data*>	each hexbyte are separated from the following by a whitespace
<hexbyte>	single data byte written in hexadecimal

Any other statement begining with a command not defined in this document shall be ignored by the parser.

' ', '\t' : Can be used as separator.

Leading and trailing separators will be ignored.

A long statement can be broken into several lines by using the character '****' at the end of each line, except for the last line.

For more details refer to the examples in B.3.

B.2 Semantics

Following is the meaning of each of the statements :

SIM Commands shall be performed in SIM mode in the INI sequence and RST. It is the default selected mode.

USIM Commands shall be performed in USIM mode in the INI sequence and RST

CMD Sends an APDU Command to the card, including (optionally) the expected response data and also (optionally) the expected status words SW1, SW2.

RST Resets and powers on the card and performs the following commands depending on the active mode.

In USIM mode: Perform a select of the first USIM application by using a select by partial AID (As specified in ETSI TS 101 220 [8])

In SIM mode: Perform a SIM select (CLA = A0) of the MF.

INI Performs the terminal profile with the following data in USIM mode if 'USIM' key word is present at the beginning of the script, or Performs the terminal profile with the following data in SIM mode otherwise (no mode specified or 'SIM' key word specified at the beginning of the script).
Afterwards, it shall perform all the fetch and terminal response commands until there is no proactive session in progress.

REM Used for comments

SWI Activates a switch condition. Every labelled list represents a list of statements to be executed, if the label matches the SW resulting from the previously executed command.

Evaluation of expected response and status in the case of a CMD:

<response> data within **[**…**]** has to be checked, it needs to be present for an outgoing command. Bytes written as XX shall not be checked by the APDU tool.

<status> status contained within **(**…**)** has to be checked; when several status are valid they shall be separated by commas. Nibble written as X shall not be checked by the APDU tool.

B.3 Example

REM this is an example

```
SIM

RST
INI FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
REM Case 1 example
CMD A0 C2 00 00 00 \
    (91 33 , 69 XX)

REM Case 2 example
CMD A0 B6 00 00 07 \
    [XX XX XX 55 55 XX 55] \
    (91 33 , 67 XX)

CMD A0 B6 00 00 07 \
    (91 33 , 67 XX)

CMD A0 C0 00 00 1F \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 28 A4 05 \
     02 0D CC CC CC CC CC CC CC CC CC CC CC CC ] \
    (90 00)

REM Case 3 example
CMD A0 C2 00 00 33 \
```

```
    D1 31 82 02 83 81 06 05 80 11 22 33 44 8B 24 40 \
    08 00 24 23 85 18 41 04 51 10 10 00 00 00 00 13 \
    02 70 00 00 0E 0D 00 00 00 00 28 A4 05 00 00 00 \
    00 00 00 \
    (90 00)

REM Case 4 example with switch statement
CMD 00 A4 04 00 10 \
    A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 02 \
    (61 XX, 6A 82)

SWI {
61 XX:
CMD 00 C0 00 00 14 \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 \
     02 02 CC CC] \
    (90 00)

CMD A0 A4 00 00 02 \
    3F 00

6A 82:
RST
}
```

B.4 Style and formatting

In order to show a common appearance all the scripts shall follow those format rules:

If the mode ('SIM' or 'USIM') is specified, it shall be the first command of the script. SIM mode is the default mode.

start always with a 'RST'.

The command, data to be checked and status to be checked shall be presented in the following order:

CMD COMMAND [EXPECTED DATA] (EXPECTED STATUS)

APDU shall be presented with command (CLA INS P1 P2 P3) in one line and data (if present) in next line grouped 16 bytes per line (see example above).

The expected data (if present) shall be presented in 16 bytes groups per line (see example above).

Annex C (normative): Default Prepersonalisation

C.1 General Default Prepersonalisation

This table shows the default prepersonalisation, the file system and the files' content, that the test (U)SIM cards shall contain unless otherwise stated.

Files under MF:

Name	Identifier	Default Value	Special Features
EF_{ICCID}	2FE2	0F FF FF FF FF FF FF FF FF FF	This value is not compliant with GSM 11.11
EF_{IMSI}	6F07	FF FF FF FF FF FF FF FF FF	This value is not compliant with GSM 11.11
EF_{LP}	6F05	01 FF FF FF	
EF_{Kc}	6F20	FF FF FF FF FF FF FF FF 07	
$EF_{PLMNsel}$	6F30	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{HPLMN}	6F31	05	
EF_{ACMmax}	6F37	00 00 00	Access condition UPDATE: CHV1
EF_{SST}	6F38	FF 3F C3 0F 0C 00 FF 0F 00 33	
EF_{ACM}	6F39	00 00 00	Access condition UPDATE: CHV1
EF_{PUCT}	6F41	FF FF FF 00 00	Access condition UPDATE: CHV1
EF_{BCCH}	6F74	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{ACC}	6F78	00 00	
EF_{FPLMN}	6F7B	FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{LOCI}	6F7E	FF FF FF FF 00 F0 00 00 00 FF 01	
EF_{AD}	6FAD	00 FF FF	
EF_{Phase}	6FAE	03	
EF_{FDN}	6F3B	Default value in all the records: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 5
EF_{SMSP}	6F42	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{LND}	6F44	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{SMSS}	6F43	FF FF	
EF_{SMS}	6F3C	1st record: 00 FF … FF(length 176) 2nd record:00 FF … FF(length 176) 3rd record: 00 FF … FF(length 176)	Records: 3
EF_{ADN}	6F3A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1

		FF	
EF_{CCP}	6F3D	FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{MSISDN}	6F40	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{SDN}	6F49	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{SUME}	6F54	85 0C 54 4F 4F 4C 4B 49 54 20 54 45 53 54 FF FF FF FF	
EF_{CBMI}	6F45	FF FF	
EF_{CBMID}	6F48	10 80	
EF_{CBMIR}	6F50	10 80 10 9F	
EF_{IMG}	4F20	FF FF FF FF FF FF FF FF FF FF FF 1st record:	

The default value for the CHV1 shall be "0x31 0x31 0x31 0x31 0xFF 0xFF 0xFF 0xFF" and its state shall be 'enabled' during tests execution.

The default value for PIN1 is the same than CHV1, so CHV1 has to be mapped on PIN1.

The maximum number of tries for the CHV1 checking shall be 3.

The default value for the Unblock CHV1 value shall be "0x33 0x33 0x33 0x33 0x33 0x33 0x33 0x33".

C.2 Sim.Access.SimView test default prepersonalisation

C.2.1 $DF_{SIMTEST}$ (SIM Test)

Identifier: '0319'

C.2.2 EF_{TNR} (Transparent Never Read)

Identifier: '6F01'		Structure: transparent		Mandatory
File size: 3 bytes		Update activity: low		
Access Conditions: READ NEVER UPDATE ALWAYS INVALIDATE ALWAYS REHABILITATE ALWAYS				
Bytes	Description	Default Value	M/O	Length
1 – 3	Test Data	AA AA AA	M	3 bytes

C.2.3 EF_{TNU} (Transparent Never Update)

Identifier: '6F02'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	NEVER
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1 - 3	Test Data	55 55 55	M	3 bytes

C.2.4 EF_{TARU} (Transparent Always Read and Update)

Identifier: '6F03'	Structure: transparent	Mandatory
File size: 260 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1 - 260	Test Data	FF ... FF	M	260 bytes

C.2.5 EF_{CNR} (Cyclic Never Read)

Identifier: '6F04'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	NEVER
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.6 EF_{CNU} (Cyclic Never Update)

Identifier: '6F05'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	NEVER
INCREASE	NEVER
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.7 EF_{CNIC} (Cyclic Never Increase)

Identifier: '6F06	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	NEVER
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.8 EF_{CNIV} (Cyclic Never Invalidate)

Identifier: '6F07	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	NEVER
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.9 EF_{CNRH} (Cyclic Never Rehabilitate)

Identifier: '6F08'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	NEVER

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.10 EF_{CARU} (Cyclic Always Read and Update)

Identifier: '6F09'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	55 55 55	M	3 bytes
2	Test Data	AA AA AA	M	3 bytes

C.2.11 EF_{LNR} (Linear Fixed Never Read)

Identifier: '6F0A'	Structure: linear fixed	Mandatory
Record length: 4 bytes	Update activity: low	

Access Conditions:

READ	NEVER
UPDATE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	FF FF FF FF	M	4 bytes
2	Test Data - Record 2	FF FF FF FF	M	4 bytes

C.2.12 EF_{LNU} (Linear Fixed Never Update)

Identifier: '6F0B'	Structure: linear fixed	Mandatory
Record length: 4 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	NEVER
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	FF FF FF FF	M	4 bytes
2	Test Data - Record 2	FF FF FF FF	M	4 bytes

C.2.13 EF_{LARU} (Linear Fixed Always Read and Update)

Identifier: '6F0C'	Structure: linear fixed	Mandatory
Record length: 4 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	55 55 55 55	M	4 bytes
2	Test Data - Record 2	AA AA AA AA	M	4 bytes

C.2.14 EF_{CINA} (Cyclic Increase Not Allowed)

Identifier: '6F0D'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS (see note 1)
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

Note 1: This file will be personalised in a way such that increase is not allowed, as indicated by the FCI byte 8, bit 7 (GSM 11.11: FCI structure of an EF returned by the SELECT command)

C.2.15 EF_{TRAC} (Transparent Read Access Condition CHV2)

Identifier: '6F0E'	Structure: transparent	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	CHV2
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

C.2.16 EF_{TIAC} (Transparent Invalidate Access Condition CHV1)

Identifier: '6F0F'	Structure: transparent	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	CHV1
REHABILITATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

C.2.17 EF_{CIAC} (Cyclic Increase Access Condition CHV2)

Identifier: '6F10'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	CHV2
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.18 EF_{CIAA} (Cyclic Increase Access Condition ADM)

Identifier: '6F11'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ADM
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.19 EF_{CNRI} (Cyclic Never Rehabilitate Invalidated)

Identifier: '6F12'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	NEVER

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

The file status shall be invalidated as defined in [2]

Annex D (normative):
Loading , testing and cleaning script examples.

See attached the file Annex_D_Examples.zip

Annex E (normative): Test Area Files

See attached files:

- Annex_E_SIM_SourceScript.zip
- Annex_E_USIM_SourceScript.zip
- Annex_E_SourceJavaCode.zip

The Annex_E_SIM_ SourceScript.zip developes the following folders tree:

SIM
- Remote
  - Test_Area_reference
    - SIM_Test_Area_reference_x.scr
    - SIM_Test_Area_reference_x.clr
    - SIM_Test_Area_reference_x.ldr
- Security
  - Test_Area_reference
    - SIM_Test_Area_reference_x.scr
    - SIM_Test_Area_reference_x.clr
    - SIM_Test_Area_reference_x.ldr

The Annex_E_USIM_ SourceScript.zip developes the following folders tree:

USIM
- Remote
- Test_Area_reference
  - USIM_Test_Area_reference_x.scr
  - USIM_Test_Area_reference_x.clr
  - USIM_Test_Area_reference_x.ldr
- Security
  - Test_Area_reference
  - USIM_Test_Area_reference_x.scr
  - USIM_Test_Area_reference_x.clr
  - USIM_Test_Area_reference_x.ldr

The Annex_E_SourceJavaCode.zip developes the following folders tree:

SIM
- Test
- Remote
  - Test_Area_reference
    - Test_Area_reference_x.java
    - Test_Area_reference_x.par
  - Security
    - Test_Area_reference
    - Test_Area_reference_x.java
    - Test_Area_reference_x.par

Note that these applets are used for both test areas (SIM and USIM).

Annex F (Normative): Configuration Parameters File

This file describes all the mandatory and optional parameters that are used in order to create the loading script(s) for one test area. The configuration parameters file contains the values for the parameters needed in order to generate the loading and cleanup scripts.

The name of the parameters file will be ***<test area reference>_<n>*.par**.

The number <n> is associated with the loading/cleanup script number, i.e. RAM_RAB_CMDS_1.par is used to generate RAM_RAB_CMDS _1.ldr etc.

F.1 Syntax

The general syntax for this file will be:

```
<file> ::= <section>+
<section> ::= <section heading> <line break> <section body>
<section heading> ::= '[' <name> ']'
<section body> ::= <parameter assignment>+
<parameter assignment> ::= <name> '=' <value> <line break>
```

Where '+' indicates one or more repetitions of the previous syntax element.

Any text included between the symbol '*;*' and the end of line is considered a comment and ignored by parsing tools.

Empty values are considered valid. They are used to indicate that an optional value is not present.

Names of sections, names of parameters and values are case-sensitive.

Blank spaces and Tabs between tokens are allowed and will be ignored by the parser.

When values represent a sequence of bytes, they are expressed in hexadecimal format, where every 2 digits represent one byte. Blank space between bytes is optional.

Example:

```
; comment


[Section1]

Parameter11 = 00 11 22 33

Parameter12 = 0101   ; another comment


[Section2]

Parameter21 = vvwwxxyyzz
```

F.2 File Contents and Organisation

Parameters in this file are organised in the following sections:

[CONVERT]	Conversion parameters used during conversion (i.e. CAP file generation)
[INSTALL(load)]	Parameters used by the Install for Load command
[LOAD]	Parameters used by the Load command
[INSTALL(install)]	Parameters used by the Install for Install command

All sections may appear only once in the file, except for the "INSTALL(install)" section. If that section appears more than once, it will apply to different applet instances, in sequence.

F.2.1 Default values, order and processing

The ordering of the parameters and the sections is relevant, since parameter names may be repeated and apply to different applets.

When one single parameter is repeated within one section, it refers to different applets. The value of the n^{th} appearance of the parameter applies to applet *n*.

When one section is repeated (INSTALL(install)), then the n^{th} appearance of the section applies to applet *n*. Parameter/value pairs which are found in one appearance of the section are valid for the subsequent applets as long as they are not overriden. For example, first INSTALL(install) may contain all values for parameters, whereas the subsequent INSTALL(install) sections may only contain parameters whose values change.

If one required parameter is missing from one section, the last defined value of this parameter in a previous section of the same file will be used.

F.2.2 CONVERT Section

These parameters allow configuration of the conversion process of the Java class file(s) into one CAP file.

Parameter	Description
PackageAID	AID of the package
PackageName	Fully qualified name of the package
PackageVersion	Version of the package
AppletClassAID	AID of the applet
AppletClassName	Name of the applet

F.2.3 INSTALL(load) Section

Here are the parameters to be included in the Install(Load) command (as specified in 3GPP TS 23.048 [6]).

Parameter	Description
PackageAID	AID of the package
PackageNonVolatileMemSize	Non Volatile memory space (in bytes) required for package loading
InstallationNonVolatileMemSize	Non volatile memory required for installation, in bytes
InstallationVolatileMemSize	Volatile memory required for installation, in bytes

F.2.4 LOAD Section

Here are the parameters to be included in the Load command (as specified in 3GPP TS 23.048 [6]).

Parameter	Description
MaxLoadCommandDataLength	Maximum length of the data provided in the load command (P3 parameter of the LOAD APDU embedded in the command packet)

F.2.5 INSTALL(install) Section

Here are the parameters to be included in the Install(Install) command (as specified in 3GPP TS 23.048 [6]

Parameter	Description
PackageAID	AID of the package
AppletClassAID	AID of the applet
InstanceAID	AID of the instance of the applet
InstallationNonVolatileMemSize	Non volatile memory required for installation, in bytes
InstallationVolatileMemSize	Volatile memory required for installation, in bytes
AccessDomain	Specify the SIM files that may be accessed by the applet and the operations allowed on these files. This parameter includes the Access Domain Parameter (ADP) and Access Domain Data (ADD)
PriorityLevel	Priority level of the Toolkit applet instance
MaxNumberOfTimers	Maximum number of timers allowed for this applet instance
MaxMenuEntryTextLength	Maximum text length for a menu entry
MaxNumberOfMenuEntries	Maximum number of menu entries allowed for this applet instance
MenuEntriesPositionIdentifier	For each menu entry: Position and identifier of that menu entry
MaxNumberOfChannels	Maximum Number of channels for this applet instance
MSLFieldLength	Length of Minimum Security Level field
MSLParameter	MSL Parameter
MSLData	MSL Data
AppletSpecificParameters	Parameters specific to the applet

The applet shall be installed with install(install and make selectable) command.

F.3 Full example

```
[CONVERT]

PackageAID = A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 00 00

PackageName = sim.test.remote.RAM_RAB_CMDS

PackageVersion = 1.0

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 10 01

AppletClassName = RAM_RAB_CMDS_1

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 20 02

AppletClassName = RAM_RAB_CMDS_2


[INSTALL(load)]

PackageNonVolatileMemSize = 0D27

;InstallationNonVolatileMemSize = 0400

;InstallationVolatileMemSize = 0000


[LOAD]

MaxLoadCommandDataLength = 6C ; max value


[INSTALL(install)]

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 10 01

InstanceAID =    A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 11 02

InstallationNonVolatileMemSize = 0400

InstallationVolatileMemSize = 0000

AccessDomain = 00

PriorityLevel = FF
```

```
MaxNumberOfTimers = 00

MaxMenuEntryTextLength = 10

MaxNumberOfMenuEntries = 01

MenuEntriesPositionIdentifier = 0001

AppletSpecificParameters =


[INSTALL(install)]

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 20 01

InstanceAID =    A0 00 00 00 30 00 02 FF FF FF FF 89 02 00 21 02

InstallationNonVolatileMemSize = 0200

InstallationVolatileMemSize = 0000

MenuEntriesPositionIdentifier = 0002

MaxNumberOfChannels = 05

MSLFieldLength = 00

MSLParameter =

MSLData =
```

```
; rest of INSTALL(install) parameters are taken from previous INSTALL(install)...
```

Annex G (normative): Specific RFM tests applicability

The following tests are only applicable for smart cards that implement a RFM application with no security level:

In Annex_E_SIM_SourceScript:

RFM_SCC_INPT

RFM_SCC_OUPT

RFM_SRB_CMDS

RFM_SRP_SMSP

All the tests in Security section (**SEC**_XXX_YYY).

In Annex_E_SSIM_SourceScript:

RFM_UCC_INPT

RFM_UCC_OUPT

RFM_URB_CMDS

RFM_URP_SMSP

All the tests in Security section (**SEC**_XXX_YYY).

The following tests are only applicable for smart cards that implement a RAM application with security level set to CC integrity:

In Annex_E_SIM_SourceScript:

SEC_SPP_SMR

All the tests with the acronyms **ANA**_XXX_YYY and **RAM**_XXX_YYY in Remote section.

In Annex_E_USIM_SourceScript:

SEC_SPP_SMR

All the tests with the acronyms **ANA**_XXX_YYY and **RAM**_XXX_YYY in Remote section.

Annex H (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
2004-12	T#26	TP-040272	-	-	-	Presented for approval	2.0.0	5.0.0
2005-01						Editorial correction to cover page	5.0.0	5.0.1
2005-10	CP#26	CP-050328	002		F	Correction to test script	5.0.1	5.1.0
		CP-050333	001		F	CR to 31.048 Rel-5: Modification of TCs 23 and 27 in test area ANA_INS_INMS		
			003		F	Correction of tests of Security Mechanism for the Response Packet		