INTERNATIONAL STANDARD

ISO/IEC 19823-10

Second edition
2020-01

Information technology — Conformance test methods for security service crypto suites —

Part 10: Crypto suite AES-128

Technologies de l'information — Méthodes d'essai de conformité pour les suites cryptographiques des services de sécurité —

Partie 10: Suite cryptographique AES-128

Reference number
ISO/IEC 19823-10:2020(E)

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

Published in Switzerland

Contents

Page

Foreword iv

Introduction v

1 Scope 1

2 Normative references 1

3 Terms, definitions, symbols and abbreviated terms 1
3.1 Terms and definitions 1
3.2 Symbols and abbreviated terms 2

4 Test methods 2
4.1 General 2
4.2 By demonstration 2
4.3 By design 2

5 Test methods with respect to the ISO/IEC 18000 series 2
5.1 Test requirements for ISO/IEC 18000-3 Interrogators and Tags 2
5.2 Test requirements for ISO/IEC 18000-63 Interrogators and Tags 3

6 Test methods with respect to the ISO/IEC 29167-10 Interrogators and Tags 3
6.1 Test map for optional features 3
6.2 Additional parameters required as input for the test 4
6.3 Crypto suite requirements 4
6.3.1 General 4
6.3.2 Crypto suite requirements of ISO/IEC 29167-10:2017, Clauses 1 to 6 5
6.3.3 Crypto suite requirements of ISO/IEC 29167-10:2017, Clauses 7 to 12 5
6.3.4 Crypto suite requirements of ISO/IEC 29167-10:2017, Annex A 21
6.3.5 Crypto suite requirements of ISO/IEC 29167-10:2017, Annex E 21

7 Test patterns 26
7.1 General 26
7.2 Test pattern information 26
7.2.1 General 26
7.2.2 Information related to ISO/IEC 18000-3 MODE 1 26
7.2.3 Information related to ISO/IEC 18000-63 27
7.3 Test pattern descriptions 27
7.3.1 General 27
7.3.2 Test pattern 01 (TAM reject message when "AuthMethod" is '11') 27
7.3.3 Test pattern 02 (TAM1 execution and error handling) 28
7.3.4 Test pattern 03 (TAM1 execution for all keys) 29
7.3.5 Test pattern 04 (TAM1 store Tag reply in the response buffer) 30
7.3.6 Test pattern 05 (TAM1 with Challenge, read Tag reply from the response buffer) 31
7.3.7 Test pattern 06 (TAM2 execution and error handling) 32
7.3.8 Test pattern 07 (TAM2 unauthorized use of KeyID for profile) 36
7.3.9 Test pattern 08 (TAM2 execution for all keys) 37
7.3.10 Test pattern 09 (MAM1 execution and error handling) 37
7.3.11 Test pattern 10 (MAM2 execution and error handling) 39
7.3.12 Test pattern 11 (MAM1 and MAM2 execution for all keys) 43

Bibliography 45

Foreword

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

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

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

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

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

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 19823-10:2017), which has been technically revised.

The main changes compared to the previous edition are as follows:

— In addition to Tag Authentication, this edition also defines support for Interrogator authentication and Mutual Authentication. This version describes the test methods for the additional functionality.

A list of all parts in the ISO/IEC 19823 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The ISO/IEC 29167 series describes security services as applicable for the ISO/IEC 18000 series. The various parts of ISO/IEC 29167 describe crypto suites that are optional extensions to the ISO/IEC 18000 series air interfaces.

The ISO/IEC 19823 series describes the conformance test methods for security service crypto suites. It is related to the ISO/IEC 18047 series, which describes the radio frequency identification device conformance test methods, in the same way as the ISO/IEC 29167 series is related to the ISO/IEC 18000 series.

These relations mean that for a product that is claimed to conform to a pair of ISO/IEC 18000 and ISO/IEC 29167 documents, then the test methods of the ISO/IEC 18047 and ISO/IEC 19823 documents apply. If a product supports more than one part of ISO/IEC 18000 or ISO/IEC 29167, all related parts of ISO/IEC 18047 and ISO/IEC 19823 apply.

NOTE 1 The conformance test requirements of ISO/IEC 18000-6, ISO/IEC 18000-61, ISO/IEC 18000-62, ISO/IEC 18000-63, ISO/IEC 18000-64 are currently all in ISO/IEC 18047-6.

This document describes the test methods for the AES-128 crypto suite as standardized in ISO/IEC 29167-10.

NOTE 2 Test methods for interrogator and tag performance are covered by the ISO/IEC 18046 series.

Information technology — Conformance test methods for security service crypto suites —

Part 10: **Crypto suite AES-128**

1 Scope

This document describes test methods for determining the conformance of security crypto suites defined in ISO/IEC 29167-10.

This document contains conformance tests for all mandatory and applicable optional functions.

The conformance parameters are the following:

— parameters that apply directly affecting system functionality and inter-operability;

— protocol including commands and replies;

— nominal values and tolerances.

Unless otherwise specified, the tests in this document are intended to be applied exclusively to RFID Tags and Interrogators defined in the ISO/IEC 15693 series and in the ISO/IEC 18000 series using ISO/IEC 29167-10.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 17025, *General requirements for the competence of testing and calibration laboratories*

ISO/IEC/TR 18047-3:2011, *Information technology — Radio frequency identification device conformance test methods — Part 3: Test methods for air interface communications at 13,56 MHz*

ISO/IEC 18047-6:2017, *Information technology — Radio frequency identification device conformance test methods — Part 6: Test methods for air interface communications at 860 MHz to 960 MHz*

ISO/IEC 19762, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

ISO/IEC 29167-10:2017, *Information technology — Automatic identification and data capture techniques — Part 10: Crypto suite AES-128 security services for air interface communications*

3 Terms, definitions, symbols and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 and ISO/IEC 29167-10 apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviated terms given in ISO/IEC 19762 apply.

4 Test methods

4.1 General

This clause describes the general test methods for ISO/IEC 29167-10. As the parts of ISO/IEC 19823 are always tested in relation with the ISO/IEC 18047 series, a duplication of information requirements and specifications should be avoided.

Clause 5 defines elements that are assumed to be covered in the respective part of the ISO/IEC 18047 series and, therefore, shall not be addressed in the ISO/IEC 19823 series. They may only be defined in the ISO/IEC 19823 series if ISO/IEC 18047 does not define them, although a revision of the respective part of the ISO/IEC 18047 series is the preferred option.

Clause 6 defines elements that are not expected to be covered by the ISO/IEC 18047 series and, therefore, shall be addressed in the respective parts of the ISO/IEC 19823 series.

4.2 By demonstration

“By demonstration” means laboratory testing of one or, if required for statistical reasons, multiple products, processes or services to ensure conformance.

A test laboratory meeting the requirements of ISO/IEC 17025 shall be selected for the performance of the indicated testing to ensure conformance of the component or system.

For protocol requirements that are verified **by demonstration**, the test conditions are specified by this document. The detailed test plan is at the discretion of the test laboratory.

4.3 By design

“By design” means design parameters and/or theoretical analysis that ensure conformance. A vendor submitting a component or system for conformance testing shall provide the necessary technical information, in the form of a technical memorandum or similar. A test laboratory shall issue a test certificate indicating whether the technical analysis was sufficient to ensure conformance of the component or system.

For protocol requirements that are verified **by design**, the method of technical analysis is at the discretion of the submitting vendor and is not specified by this document. In general, the technical analysis shall have sufficient rigor and technical depth to convince a test engineer knowledgeable of the protocol that the particular requirement has been met.

5 Test methods with respect to the ISO/IEC 18000 series

5.1 Test requirements for ISO/IEC 18000-3 Interrogators and Tags

The following mandatory requirements and applicable optional requirements of ISO/IEC TR 18047-3:2011 shall be fulfilled:

- 5.2 Default conditions applicable to the test methods

Before a DUT is tested according to this document, it shall successfully pass the following prerequisite from ISO/IEC TR 18047-3:2011:

- 5.3 Conformance tests for ISO/IEC 18000-3 Mode 1

5.2 Test requirements for ISO/IEC 18000-63 Interrogators and Tags

The following mandatory requirements and applicable optional requirements of ISO/IEC 18047-6:2017 shall be fulfilled:

- Clause 4 Default conditions applicable to the test methods
- Clause 5 Set up of test equipment

Before a DUT is tested according to this document, it shall successfully pass the following prerequisite from ISO/IEC 18047-6:2017:

- Clause 8 Conformance tests for ISO/IEC 18000-63

6 Test methods with respect to the ISO/IEC 29167-10 Interrogators and Tags

6.1 Test map for optional features

Table 1 lists all optional features of this crypto suite and shall be used as a template to report the test results.

Table 1 — Test map for optional features

#	Feature	Additional requirements	Mark items to be tested for supplied product	Test results
1	TAM2	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
1.1	Memory profiles and MPI	Shall be tested for all the declared memory profiles and for every supported key.		
		MAX_Profiles=Number of memory profiles.		
		MAX_KeyID=Number of keys supported.		
1.21	ProtMode=0000_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
1.22	ProtMode=0001_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
1.23	ProtMode=0010_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
1.24	ProtMode=0011_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
2	IAM1	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		

Table 1 *(continued)*

#	Feature	Additional requirements	Mark items to be tested for supplied product	Test results
3	IAM2	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
4	IAM3	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
5.1	Memory profiles and MPI	Shall be tested for all the declared memory profiles and for every supported key.		
		MAX_Profiles=Number of memory profiles.		
		MAX_KeyID=Number of keys supported.		
5.21	ProtMode=0000_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
5.22	ProtMode=0001_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
5.23	ProtMode=0010_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
5.24	ProtMode=0011_b	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
6	MAM1	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		
7	MAM2	Shall be tested with the *Authenticate* command of the relevant part of ISO/IEC 15693 or ISO/IEC 18000.		

Table 3 lists all crypto suite requirements that shall be tested in dependence of the features of Table 1 as supported by the DUT. Items marked with M are mandatory and shall be tested for each DUT.

6.2 Additional parameters required as input for the test

Table 2 lists all additional test parameters of this crypto suite.

Table 2 — Additional test parameters

#	Feature	Additional requirement	Value
1	Maximum BlockSize	Shall be provided to ensure that only test results for supported parameters are taken into consideration.	
2	TAM2 Revision	Shall be provided to ensure that only test results for supported parameters are taken into consideration.	0 or 1

6.3 Crypto suite requirements

6.3.1 General

This clause contains all requirements of ISO/IEC 29167-10.

6.3.2 Crypto suite requirements of ISO/IEC 29167-10:2017, Clauses 1 to 6

All the requirements of ISO/IEC 29167-10:2017, Clauses 1 to 6 are mandatory, inherently by design only.

6.3.3 Crypto suite requirements of ISO/IEC 29167-10:2017, Clauses 7 to 12

Table 3 contains all requirements of ISO/IEC 29167-10:2017, Clauses 7 to 12.

Table 3 — Crypto suite requirements of ISO/IEC 29167-10:2017, Clauses 7 to 12

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0020	7 Crypto suite state diagram	The Tag shall transition from the Start State to the Next State conforming to the requirements specified in Annex A.	M	Tag	By design
0030	8 Initialization and resetting	After power-up and after a reset, the crypto suite shall transition into the **Initial** state.	M	Tag	By design
0040	8	After the Tag encounters an error condition, it shall transition into the **Initial** state.	M	Tag	By design
0050	8	After the Tag encounters an error condition, it may send an error reply to the Interrogator, but in that case the Tag shall select one Error Condition from the list that is specified in Annex B.	M	Tag	By design
0060	8	A transition to **Initial** state shall also cause a reset of all variables used by the crypto suite.	M	Tag	By design
0070	8	Implementations of this crypto suite shall assure that all memory used for intermediate results is cleared after each operation (message-response pair) and after reset.	M	Tag	By design
0080	9.2 Adding custom data	The authentication message shall include the reference KeyID to select an encryption key in Table 27 (see Clause 11).	M	Interrogator	By design
0090	9.2	If protection of integrity and authenticity of the data is requested, the selected reference KeyID shall also contain a MAC key.	M	Interrogator	By design
0100	9.2	A Tag that supports including custom data in the authentication process shall define at least one and at most 16 memory profiles.	M	Tag	By demonstration using test pattern 08
0110	9.2	The memory profiles may also be linked to a key in Table 27 that shall be used for the encryption process to protect the data.	M	Tag	By demonstration using test pattern 07
0120	9.2	The custom data block shall be defined by the parameters BlockSize, Profile, Offset and BlockCount.	M	Interrogator / Tag	By design
0130	9.2	The mode of operation that shall be used for the encryption and/or protection of the custom data is specified by ProtMode.	M	Interrogator / Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0140	9.2	BlockSize shall select the size of the custom data block; "0_b" specifies custom data in 64-bit blocks, "1_b" specifies custom data as 16-bit blocks.	M	Interrogator / Tag	By design
0150	9.2	Profile shall select one of the memory profiles that are supported by the Tag. The memory profiles are specified in Annex E.	M	Interrogator / Tag	By design
0160	9.2	Maximum binary value is "1111_b", or decimal 15, corresponding to a maximum number of 16 blocks of custom data that shall be included.	M	Tag	By design
0170	9.2	If the number of included bits of the custom data including the header is not a multiple of 128, then padding with zeroes shall be applied to the least significant bits of the last block that has a non-zero block size of less than 128 bits.	M	Tag	By design
0180	9.2	The Interrogator shall maintain the value of BlockCount for use as part of the MAC verification process.	M	Interrogator	By design
0190	9.2	The Tag manufacturer shall specify the number of custom data blocks that can be included.	M		By design
0200	9.2	The minimum value of *D* shall be 1. The maximum value of *D* supported by the Tag is specified by the Tag manufacturer.	M		By design
0210	9.2	ProtMode specifies the mode of operation that shall be used for the encryption and/or protection of the custom data.	M	Interrogator / Tag	By design
0220	9.3 Message and response formatting	The crypto suite shall parse the Messages and process the data based on the value of AuthMethod, which is the first parameter (first two bits) of all Messages.	M	Tag	By design
0230	9.3	The Messages for Tag Authentication, Interrogator Authentication and Mutual Authentication shall be distinguished by AuthMethod.	M	Interrogator / Tag	By design
0240	9.3	If AuthMethod = "00_b", the Tag shall parse the Message for Tag Authentication as described in 9.4.	M	Tag	By design
0250	9.3	If AuthMethod = "01_b", the Tag shall parse Message for Interrogator Authentication as described in 9.5.	M	Tag	By design
0260	9.3	If AuthMethod = "10_b", the Tag shall parse Message for Mutual Authentication as described in 9.6.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0270	9.3	If AuthMethod = "11_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration, using the test pattern 01
0280	9.4.1 TAM	If CustomData = "0_b", the Tag shall parse the TAM1 Message for Tag Authentication without custom data as described in 9.4.2.	M	Tag	By demonstration, using the test pattern 03
0280	9.4.1 TAM	If CustomData = "1_b", the Tag shall parse the TAM2 Message for Tag Authentication with custom data as described in 9.4.5.	M	Tag	By demonstration, using the test pattern 08
0280	9.4.2 TAM1	For Tag authentication, the Interrogator shall generate an 80-bit random TAM1 Interrogator challenge and include that in the TAM1 message. The TAM1 message shall also include the reference KeyID to select an encryption key in Table 27 (see Clause 11). KeyID: 8-bit value that specifies the key that shall be used for TAM1.	M	Interrogator / Tag	By demonstration, using the test pattern 03
0310	9.4.2	The Tag shall accept this message in any state. If the value of the parameters of the message is invalid, then the Tag shall transition to the **Initial** state, thereby aborting any cryptographic protocol that has not yet been completed.	M	Tag	By design
0330	9.4.2	If the length of the TAM1 message is <> 96 bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By demonstration, using the test pattern 02
0340	9.4.2	If TAM1_RFU[4:0] is <> "00000_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration, using the test pattern 02
0350	9.4.2	If the Tag does not support key[KeyID].*ENC_key*, then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 02 (test pattern 5)
0360	9.4.3	If all parameters have been successful verified, then the Tag shall generate a response as specified in Table 5. The Tag shall generate the random data *TRnd_TAM1*[31:0] and encrypt the concatenation of the constant C_TAM1[15:0], the random data *TRnd_TAM1*[31:0] and the challenge IChallenge_TAM1[79:0] using Key[KeyID].*ENC_key*.	M	Tag	By demonstration using test pattern 03
0380	9.4.3	After returning the TAM1 Response (TResponse), the Tag shall remain in the **Initial** state.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0390	9.4.4	The Interrogator (or the external application controlling the Interrogator) decrypts the TAM1 Response (TResponse) and shall verify whether C_TAM1 and IChallenge_TAM1 have the correct value.	M	Interrogator	By demonstration using test pattern 03
0400	9.4.5 TAM2 Message	The Interrogator shall generate an 80-bit random number for use as TAM2 Interrogator challenge.	M	Interrogator	By design
0410	9.4.5	BlockCount[3:0]: number that defines the size of the custom data as a number of 16-bit or 64-bit blocks. If the number of included bits of the custom data including header is not a multiple of 128, then padding with zeroes shall be applied to the least significant bits of the last block that has a non-zero block size of less than 128 bits.	M	Interrogator	By design
0420	9.4.5	The Interrogator shall maintain the value of BlockCount for use as part of the MAC verification process.	M	Interrogator	By design
0430	9.4.5	The Tag manufacturer shall specify the number of custom data blocks that can be included.	M	Tag	By design
0440	9.4.5	ProtMode[3:0]: value to select the mode of operation that shall be used to process the custom data as specified in Table 3.	M	Interrogator	By design
0450	9.4.5	The Tag shall accept this message in any state.	M	Tag	By design
0460	9.4.5	If the parameters of the message are invalid, then the Tag shall transition to the **Initial** state, thereby aborting any cryptographic protocol that has not yet been completed.	M	Tag	By design
0470	9.4.5	If the length of the TAM2 message is <> 120 bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By demonstration using test pattern 06
0480	9.4.5	If BlockSize = "1_b" and the Tag does not support value "1_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
0490	9.4.5	If TAM2_Rev specifies a TAM2 message format that is not supported by the Tag, then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
0500	9.4.5	If TAM2_RFU[2:0] is <> "000_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 06
0510	9.4.5	If the Tag does not support key[KeyID].*ENC_key*, then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 06

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0520	9.4.5	If the memory profile specified in Profile is not supported by the Tag, then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 06
0530	9.4.5	The Tag shall check if the specified memory profile has the right to use KeyID for further processing: else key[KeyID] is not authorized for this memory profile and the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 07
0550	9.4.5	If the block of custom data specified by BlockSize, Profile, Offset and BlockCount is not supported by the Tag, then the Tag shall return a "Memory Overrun" error condition.	M	Tag	By demonstration using test pattern 06
0560	9.4.5	If the ProtMode value is not supported by the Tag, then the Tag shall return a “Not Supported” error condition.	M	Tag	By design
0570	9.4.6.1 TAM2 Response	If all parameters have been successfully verified, then the Tag shall proceed with parsing the TAM2 message.	M	Tag	By demonstration using test pattern 08
0580	9.4.6.1	After returning the TAM2 Response (TResponse), the Tag shall remain in the **Initial** state.	M	Tag	By design
0590	9.4.6.2 TAM2_Rev = "0_b" and ProtMode = "0000_b"	The Tag shall add custom data in plaintext to the authentication block and generate a response as specified in Table 7.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0000_b"
0600	9.4.6.3 TAM2_Rev = "0_b" and ProtMode = "0001_b"	The Tag shall add custom data with confidentiality protection to the authentication block and generate a response as specified in Table 8. The Tag shall use AES encryption in CBC mode to encrypt all *D* custom data blocks.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0001_b"
0620	9.4.6.4 TAM2_Rev = "0_b" and ProtMode = "0010_b"	The Tag shall add custom data with integrity protection to the authentication block and generate a response as specified in Table 9. The Tag shall use AES-CMAC-96 to calculate the truncated 96-bit CMAC over the authentication block and the *D* following plaintext custom data blocks.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0010_b"

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0640	9.4.6.5 TAM2_Rev = "0_b" and ProtMode = "0011_b"	The Tag shall add custom data with confidentiality and integrity protection to the authentication block and generate a response as specified in Table 10. The Tag shall use AES encryption in CBC mode to encrypt the initial authentication block and all following *D* custom data blocks. The Tag shall use AES-CMAC-96 to calculate the truncated 96-bit CMAC over the authentication block and the *D* following encrypted custom data blocks.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0011_b"
0670	9.4.6.6 TAM2_Rev = "1_b" and ProtMode = "0000_b"	The Tag shall compute the authentication block as the encryption of *C_TAM2_0*[15:0], *TRnd_TAM2*[31:0] and IChallenge_TAM2[79:0]. The Tag shall add the header and the custom data in plaintext to the authentication block and generate a response as specified in Table 11.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0000_b"
0690	9.4.6.7 TAM2_Rev = "1_b" and ProtMode = "0001_b"	The Tag shall compute the authentication block as the encryption of *C_TAM2_1*[15:0], *TRnd_TAM2*[31:0] and IChallenge_TAM2[79:0]. The Tag shall add the header and the custom data with confidentiality protection to the authentication block and generate a response as specified in Table 12. The Tag shall use AES encryption in CBC mode to encrypt all *D* data blocks composed of the header and the custom data.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0001_b"
0720	9.4.6.8 TAM2_Rev = "1_b" and ProtMode = "0010_b"	The Tag shall compute the authentication block as the encryption of *C_TAM2_2*[15:0], *TRnd_TAM2*[31:0] and IChallenge_TAM2[79:0]. The Tag shall add the header and the custom data with integrity protection to the authentication block and generate a response as specified in Table 13. The Tag shall use AES-CMAC-96 to calculate the truncated 96-bit CMAC over the authentication block and the *D* following plaintext data blocks composed of the header and the custom data.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0010_b"

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0750	9.4.6.9 TAM2_Rev = "1_b" and ProtMode = "0011_b"	The Tag shall add the header and the custom data with confidentiality and integrity protection to the authentication block and generate a response as specified below and in Table 14. The Tag shall use AES encryption in CBC mode to encrypt the initial authentication block and all following *D* data blocks composed of the header and the custom data. The Tag shall use AES-CMAC-96 to calculate the truncated 96-bit CMAC over the authentication block and the *D* following encrypted custom data blocks.	O	Tag	By demonstration using test pattern 08, with profile that is supported by the Tag and ProtMode = "0011_b"
0780	9.4.7.2 TAM2 Final Interrogator processing TAM2_Rev = "0_b"	The Interrogator (or the external application controlling the Interrogator) decrypts the TAM2 Response (TResponse) and shall verify whether *C_TAM2* and IChallenge_TAM2 have the correct value.	M	Interrogator	By demonstration using test pattern 08 and verifying that the interrogator aborts if the local key at the interrogator is changed to a value different from that in the Tag
0790	9.4.7.3 TAM2 Final Interrogator processing TAM2_Rev = "1_b"	The Interrogator (or the external application controlling the Interrogator) decrypts the first block of TAM2 Response (TResponse) and shall verify whether *C_TAM2 constant* and IChallenge_TAM2 have the correct value.	M	Interrogator	By demonstration using test pattern 08 and verifying that the interrogator aborts if the local key at the interrogator is changed to a value different from that in the Tag
0800	9.4.7.3	If ProtMode = 0000_b, C_TAM2 shall be C_TAM2_0.	M	Interrogator	By design
0810	9.4.7.3	If ProtMode = 0001_b, C_TAM2 shall be C_TAM2_1.	M	Interrogator	By design
0820	9.4.7.3	If ProtMode = 0010_b, C_TAM2 shall be C_TAM2_2.	M	Interrogator	By design
0830	9.4.7.3	If ProtMode = 0011_b, C_TAM2 shall be C_TAM2_3	M	Interrogator	By design
0840	9.5.1 IAM	If Step = "00_b", the Tag shall parse the IAM1 Message for Interrogator Authentication as described in 9.5.2.	M	Tag	By design
0850	9.5.1 IAM	If Step = "01_b", the Tag shall parse the IAM2 and IAM3 Messages and process the data based on the value of CustomData, which is the third parameter in the IAM2 and IAM3 Messages.	M	Tag	By design
0860	9.5.1 IAM	If Step = "01_b" and CustomData = "0_b", the Tag shall parse the IAM2 Message for Interrogator Authentication without custom data as described in 9.5.5.	M	Tag	By design
0870	9.5.1 IAM	If Step = "01_b" and CustomData = "1_b", the Tag shall parse the IAM3 Message for Interrogator Authentication with custom data as described in 9.5.8.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
0880	9.5.1 IAM	If Step = "10_b", the Tag shall return a "Not Supported" error condition.	M	Tag	By design
0890	9.5.1 IAM	If Step = "11_b", the Tag shall return a "Not Supported" error condition.	M	Tag	By design
0900	9.5.2 IAM1	The Tag shall accept this message only in the **Initial** or the **IA-OK** state (unless occupied by internal processing and not capable of receiving messages).	M	Tag	By design
0910	9.5.2 IAM1	If the parameters of the message are invalid, then the Tag shall transition to the **Initial** state, thereby aborting any cryptographic protocol that has not yet been completed.	M	Tag	By design
0920	9.5.2 IAM1	If the length of the IAM1 message is <> 16 bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By design
0930	9.5.2 IAM1	If the value of IAM1_RFU[3:0] is <> "0000_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
0940	9.5.2 IAM1	If the Tag does not support key[KeyID].*ENC_key*, then it shall return a "Not Supported" error condition.	M	Tag	By design
0950	9.5.3 IAM1 Response	The Tag shall generate a random challenge *TChallenge_IAM1*[79:0] and store a copy of *TChallenge_IAM1* for subsequent verification (see 9.5.5 or 9.5.8).	M	Tag	By design
0960	9.5.3	The Tag shall store a copy of KeyID for use in 9.5.5 or 9.5.8.	M	Tag	By design
0970	9.5.3	The Tag shall send the challenge *TChallenge_IAM1* in the IAM1 Response as specified in Table 16.	M	Tag	By design
0980	9.5.3	After returning the IAM1 Response (TResponse), the Tag shall transition to the **IAM-Init** state.	M	Tag	By design
0990	9.5.4 Final Interrogator processing IAM1	The Interrogator (or the external application controlling the Interrogator) shall decrypt a concatenation of *C_IAM2* ($DA8_h$), *Purpose_IAM2*[3:0], *IRnd_IAM2*[31:0] and *TChallenge_IAM1* as input for the IAM2 Message or IAM3 Message.	M	Interrogator	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1000	9.5.5 IAM2 Message	The Tag shall accept this message only in the **IAM-Init** state (unless occupied by internal processing and not capable of receiving messages). If the Tag is not in the **IAM-Init** state, it shall abort any cryptographic protocol that has not yet been completed and shall transition to the **Initial** state.	M	Tag	By design
1030	9.5.5	If the length of the IAM2 message is <> 136 bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By design
1040	9.5.5	If the value of IAM2_RFU[2:0] is <> "000_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1050	9.5.5	If the parameter verifications have been completed successfully, the Tag shall perform an AES encryption of IResponse and retrieve *C_IAM2*[11:0], *Purpose_IAM2*[3:0], *IRnd_IAM2*[31:0] and *TChallenge_IAM1*[79:0]) for further verification.	M	Tag	By design
1060	9.5.5	Cryptographic errors shall only be returned after all checks have been completed.	M	Tag	By design
1070	9.5.5	If the value of *C_IAM2*[11:0] is <> "$DA8_h$", then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1080	9.5.5	If the value of *Purpose_IAM2*[3:0] is <> "0000_b" and not supported by the Tag, then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1090	9.5.5	If the value for *TChallenge_IAM1*[79:0] is not equal to the copy of *TChallenge_IAM1*[79:0] that has been stored in IAM1 (see 9.5.3), then the Tag shall return a "Cryptographic Error" error condition.	M	Tag	By design
1100	9.5.6 IAM2 Response	If the Interrogator Authentication has been completed successfully, the Tag shall respond with an IAM2 Response that shall be empty (zero bits).	M	Tag	By design
1110	9.5.6	After returning the IAM2 Response (TResponse), the Tag shall transition to the **IA-OK** state.	M	Tag	By design
1120	9.5.8.1 IAM3 Message	The Interrogator shall use IAM3 if it wants to write custom data in the Tag's memory using Interrogator Authentication.	M	Interrogator	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1130	9.5.8.1	If ciphertext is required, the Interrogator shall use AES in CBC decryption mode on the custom data, using the *Authentication-Block* as the Initialization Vector.	M	Interrogator	By design
1140	9.5.8.1	If required, the interrogator shall use AES-CMAC-96 to protect the integrity of the message by calculating a message authentication code over the authentication block and the following *D* custom data blocks.	M	Interrogator	By design
1150	9.5.8.1	If the number of included bits of the header and custom data is not a multiple of 128, then padding with zeroes shall be applied to the least significant bits of the last block that has a non-zero block size of less than 128 bits.	M	Interrogator	By design
1160	9.5.8.1	The Interrogator shall maintain the value of BlockCount for use as part of the MAC verification process.	M	Interrogator	By design
1170	9.5.8.1	The Tag manufacturer shall specify the number of custom data blocks that can be included.	M		By design
1180	9.5.8.1	ProtMode[3:0]: value to select the mode of operation that shall be used to process the custom data as specified in Table 3.	M	Interrogator / Tag	By design
1190	9.5.8.1	The Tag shall accept this message only in the **IAM-Init** state (unless occupied by internal processing and not capable of receiving messages).	M	Tag	By design
1200	9.5.8.1	If the Tag is not in the **IAM-Init** state, it shall abort any cryptographic protocol that has not yet been completed and shall transition to the **Initial** state.	M	Tag	By design
1210	9.5.8.1	The Tag shall verify the length of the IAM3 message.	M	Tag	By design
1220	9.5.8.1	If ProtMode is "0000_b" or "0001_b" and the length of the IAM3 message is <> (32 + 128 + *D**128) bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By design
1230	9.5.8.1	If ProtMode is "0010_b" or "0011_b" and the length of the IAM3 message is <> (32 + 128 + *D**128 + 96) bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By design
1240	9.5.8.1	If the ProtMode value is not supported by the Tag, then the Tag shall return a “Not Supported” error condition.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1250	9.5.8.1	If the value of IAM3_RFU[1:0] is <> "00$_b$", then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1260	9.5.8.1	If the memory profile specified in Profile is not supported by the Tag, then the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1270	9.5.8.1	The Tag shall check if the specified memory profile has the right to use KeyID for further processing.	M	Tag	By design
1280	9.5.8.1	[...] else key[KeyID] is not authorized for this memory profile and the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1290	9.5.8.1	If the block of custom data specified by Profile, BlockSize, Offset and BlockCount is not supported by the Tag, then the Tag shall return a "Memory Overrun" error condition.	M	Tag	By design
1300	9.5.8.1	If the block of custom data specified by Profile, BlockSize, Offset and BlockCount is write locked, then the Tag shall return a "Memory Write Error" error condition.	M	Tag	By design
1310	9.5.8.1	If the verifications have been completed successfully, the Tag shall perform an AES encryption of the authentication block in IResponse[*LoI-1:LoI-128*] and retrieve *C_IAM3*[11:0], *Purpose_IAM3*[3:0], *IRnd_IAM3*[31:0] and *TChallenge_IAM1*[79:0]) for further verification.	M	Tag	By design
1320	9.5.8.1	If ProtMode is "0000$_b$", the Tag shall check if the value of C_IAM3[11:0] is equal to C_IAM3_0.	M	Tag	By design
1330	9.5.8.1	In case of mismatch, the Tag shall return a Cryptographic Error" error condition.	M	Tag	By design
1340	9.5.8.1	If ProtMode is "0001$_b$", the Tag shall check if the value of C_IAM3[11:0] is equal to C_IAM3_1.	M	Tag	By design
1350	9.5.8.1	In case of mismatch, the Tag shall return a Cryptographic Error" error condition.	M	Tag	By design
1360	9.5.8.1	If ProtMode is "0010$_b$", the Tag shall check if the value of C_IAM3[11:0] is equal to C_IAM3_2.	M	Tag	By design
1370	9.5.8.1	In case of mismatch, the Tag shall return a Cryptographic Error" error condition.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1380	9.5.8.1	If ProtMode is "0011_b", the Tag shall check if the value of C_IAM3[11:0] is equal to C_IAM3_3.	M	Tag	By design
1390	9.5.8.1	In case of mismatch, the Tag shall return a Cryptographic Error” error condition.	M	Tag	By design
1400	9.5.8.1	If the value of *Purpose_IAM3*[3:0] is <> "0000_b" and not supported by the Tag, then the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By design
1410	9.5.8.1	If the value for *TChallenge_IAM1*[79:0] is not equal to the copy of *TChallenge_IAM1*[79:0] that has been stored in IAM1 (see 9.5.3), then the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By design
1420	9.5.8.1	If all verifications have been completed successfully, the Tag shall further process IResponse based on the value of ProtMode.	M	Tag	By design
1430	9.5.8.1	If ProtMode is "0000_b", the Tag shall process IResponse as described in 9.5.8.2.	M	Tag	By design
1440	9.5.8.1	If ProtMode is "0001_b", the Tag shall process IResponse as described in 9.5.8.3.	M	Tag	By design
1450	9.5.8.1	If ProtMode is "0010_b", the Tag shall process IResponse as described in 9.5.8.4.	M	Tag	By design
1460	9.5.8.1	If ProtMode is "0011_b", the Tag shall process IResponse as described in 9.5.8.5.	M	Tag	By design
1470	9.5.8.2	The tag shall retrieve HEADER from the IResponse[H:0] and verify that it is valid.	M	Tag	By design
1480	9.5.8.2	In case of mismatch, the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By design
1490	9.5.8.2	The Tag shall retrieve the custom data from IResponse[(D*128-H-1):0] and store it in *CUSTOMDATA*-(D*128-H).	M	Tag	By design
1500	9.5.8.3	The Tag shall recover the header and the custom data by encrypting CBC$_{ENC}$_AES (IV= IResponse[(*LoI*-1): (*LoI*-128)], Key[KeyID].*ENC_key*, IResponse[(D*128-1):0]).	M	Tag	By design
1510	9.5.8.3	Then the Tag shall retrieve HEADER from the previous result [H:0] and verify that it is valid.	M	Tag	By design
1520	9.5.8.3	In case of mismatch, the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1530	9.5.8.4	The Tag shall use AES-CMAC-96(Key[KeyID]. *MAC_key*, IResponse[(*LoI*-1):96]) to calculate the truncated 96-bit CMAC over *the authentication block* and the plaintext custom data HEADER(*H*) || *CUSTOMDATA*(D*128-H).	M	Tag	By design
1540	9.5.8.4	The Tag shall compare the result with IResponse[95:0] and return a “Cryptographic Error” error condition if the values are not identical.	M	Tag	By design
1550	9.5.8.4	The Tag shall retrieve the data from IResponse[(D*128+95):96].	M	Tag	By design
1560	9.5.8.4	From this data, the Tag shall extract HEADER and verify it.	M	Tag	By design
1570	9.5.8.4	In case of mismatch, the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By design
1580	9.5.8.4	Finally, the Tag shall extract the custom data and store it in *CUSTOMDATA*(D*128-H).	M	Tag	By design
1590	9.5.8.5	The Tag shall use AES-CMAC-96(Key[KeyID].*MAC_key*, IResponse[(*LoI*-1):96]) to calculate the truncated 96-bit CMAC over *the authentication block* and the encrypted data HEADER(*H*) || *CUSTOMDATA*(D*128-H).	M	Tag	By design
1600	9.5.8.5	The Tag shall compare the result with IResponse[95:0] and return a “Cryptographic Error” error condition if the values are not identical.	M	Tag	By design
1610	9.5.8.5	The Tag shall recover the data by encrypting CBC$_{ENC}$_AES$_{INV}$ (IV= IResponse[(*LoI*-1): (*LoI*-128)], Key[KeyID].*ENC_key*, IResponse[(D*128+95):96].	M	Tag	By design
1620	9.5.8.5	From the previous result, the Tag shall extract HEADER and *CUSTOMDATA*(D*128-H).	M	Tag	By design
1630	9.5.8.5	Then the Tag shall verify the HEADER.	M	Tag	By design
1640	9.5.8.5	In case of mismatch, the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By design
1650	9.5.8.5	If HEADER is valid, the Tag shall store the custom data in *CUSTOMDATA*(D*128-H).	M	Tag	By design
1660	9.5.9 IAM3 Response	If the Interrogator Authentication and the verification of the custom data has been completed successfully, the Tag shall write the value *CUSTOMDATA*(D*128-H), as specified by the parameters BlockSize, Profile, Offset and BlockCount, to the Tag’s memory.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1670	9.5.9	If writing the custom data *CUSTOMDATA*(*D**128-*H*) to the specified memory area results in an error, then the Tag shall return the “Memory Write Error” error condition.	M	Tag	By design
1680	9.5.9	The Tag shall respond with an IAM3 Response that shall be empty (zero bits).	M	Tag	By design
1690	9.5.9	After sending the IAM3 Response, the Tag shall transition to the **IA-OK** state.	M	Tag	By design
1700	9.6.1 MAM	If Step = "00_b", the Tag shall parse the MAM1 Message as described in 9.6.2.	M	Tag	By design
1710	9.6.1 MAM	If Step = "01_b", the Tag shall parse the MAM2 Message as described in 9.6.5.	M	Tag	By design
1720	9.6.1 MAM	If Step = "10_b", the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1730	9.6.1 MAM	If Step = "11_b", the Tag shall return a "Not Supported" error condition.	M	Tag	By design
1740	9.6.2 MAM1	The Interrogator shall generate an 80-bit random number for use as IChallenge_MAM1. To initiate the mutual authentication, the Interrogator sends a request to get a challenge from the Tag.	M	Interrogator	By design
1750	9.6.2 MAM1	The Tag shall accept this message only in the **Initial** or the **IA-OK** state (unless occupied by internal processing and not capable of receiving messages).	M	Tag	By design
1760	9.6.2 MAM1	If the parameters of the message are invalid, then the Tag shall transition to the **Initial** state, thereby aborting any cryptographic protocol that has not yet been completed.	M	Tag	By design
1770	9.6.2 MAM1	If the length of the MAM1 message is <> 96 bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By demonstration using test pattern 09
1780	9.6.2 MAM1	If the value of MAM1_RFU[3:0] is <> "0000_b", then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 09
1790	9.6.2 MAM1	If the Tag does not support key[KeyID].*ENC_key*, then it shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 09
1800	9.6.3 MAM1	The Tag shall store a copy of IChallenge_MAM1[31:0] for subsequent verification (see 9.6.5).	M	Tag	By design
1810	9.6.3 MAM1	The Tag shall store a copy of KeyID for use in 9.6.5.	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1820	9.6.3 MAM1	The Tag shall generate a random challenge *TChallenge_MAM1*[79:0] and store a copy of *TChallenge_MAM1*[79:0] for subsequent verification (see 9.6.5).	M	Tag	By design
1830	9.6.3 MAM1	The Tag shall encrypt a concatenation of the constant *C_MAM1*(DA83$_h$), *TChallenge_MAM1*[31:0] and IChallenge_MAM1[79:0] using Key[KeyID].*ENC_key* and concatenate *TChallenge_MAM1*[79:32] to the result.	M	Tag	By demonstration using test pattern 09
1840	9.6.3 MAM1	After returning MAM1 Response (TResponse), the Tag shall transition to the **MAM-Init** state.	M	Tag	By design
1850	9.6.4 MAM1	The Interrogator (or the external application controlling the Interrogator) decrypts the MAM1 Response (TResponse) and shall verify whether *C_MAM1* and IChallenge_MAM1 have the correct value.	M	Interrogator	By design
1860	9.6.5 MAM2	The Tag shall accept this message only in the **MAM-Init** state (unless occupied by internal processing and not capable of receiving messages).	M	Tag	By design
1870	9.6.5	If the Tag is not in the **MAM-Init** state, it shall abort any cryptographic protocol that has not yet been completed and shall transition to the **Initial** state.	M	Tag	By design
1880	9.6.5	If the length of the MAM2 message is <> 136 bits, then the Tag shall return an "Other Error" error condition.	M	Tag	By demonstration using test pattern 10
1890	9.6.5	If the value of MAM2_RFU[2:0] is <> "000$_b$", then the Tag shall return a "Not Supported" error condition.	M	Tag	By demonstration using test pattern 10
1900	9.6.5	If the verification of MAM2_RFU is completed, the Tag shall encrypt the Interrogator message IResponse[127:0] to retrieve *C_MAM2*[11:0], *Purpose_MAM2*[3:0], IChallenge_MAM1[31:0] and *TChallenge_MAM1*[79:0]).	M	Tag	By design
1920	9.6.5	Cryptographic errors shall only be returned after all checks have been completed.	M	Tag	By design
1930	9.6.5	If the value of *C_MAM2*[11:0] is <> "DA8$_h$", then the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By demonstration using test pattern 10
1940	9.6.5	If the value of *Purpose_MAM2*[3:0] is <> "0000$_b$" and not supported, then the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By demonstration using test pattern 10

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
1950	9.6.5	If the value for IChallenge_MAM1[31:0] is not equal to the copy of IChallenge_MAM1[31:0] that has been stored in 9.6.3, then the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By demonstration using test pattern 10
1960	9.6.5	If the value for *TChallenge_MAM1*[79:0] is not equal to the copy of *TChallenge_MAM1*[79:0] that has been stored in 9.6.3, then the Tag shall return a “Cryptographic Error” error condition.	M	Tag	By demonstration using test pattern 10
1970	9.6.6	If the Mutual Authentication has been completed successfully, the Tag shall respond with an MAM2 Response that shall be empty (zero bits).	M	Tag	By demonstration using test pattern 11
1980	9.6.6	After returning the MAM2 Response (TResponse), the Tag shall transition to the **IA-OK** state.	M	Tag	By design
1990	11	A Tag shall store one or more keys in the Key Table as specified in Table 27.	M	Tag	By design
2000	11	KeyID shall start with "00_h" and increment with one for every next key in the Key Table.	M	Interrogator/ Tag	By design
2020	11	Each key shall contain an encryption key (*ENC_key*).	M	Tag	By design
2030	11	Encryption keys shall be exclusively used for Tag authentication, Interrogator authentication, Mutual authentication and encryption of custom data.	M	Interrogator/ Tag	By design
2040	11	Message authentication keys shall be exclusively used for the authentication of custom data.	M	Interrogator/ Tag	By design
2050	11	The Tag shall maintain a record in the Key Table for each key.	M	Tag	By design
2060	11	A record of the Key Management Table is specified in Table 27 and shall have parameters for every key.	M	Tag	By design
2070	11	If the value of an MPI bit is "0_b", this key shall not be used by the related profile.	M	Tag	By design
2080	11	The MPI bit or bits for a non-existing profile on a Tag shall be permalocked to zero (bit "0_b") by the Tag manufacturer.	M	Tag	By design
2090	11	MPI is an optional parameter, but it shall be supported if the Tag supports the TAM2 mode (with custom data).	M	Tag	By design

Table 3 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
3000	11	The size and initial values in Table 27 and its mapping to their respective physical memory locations on the Tag shall be defined by the manufacturer.	M	Tag	By design

[a] All references are to ISO/IEC 29167-10.

[b] M: mandatory; items marked with "M" are mandatory and shall be tested for all devices.

O: optional; items marked with "O" are optional and shall be tested only for devices that support the feature that is indicated by the requirement.

[c] This column may define test patterns that are used for verification by demonstration.

6.3.4 Crypto suite requirements of ISO/IEC 29167-10:2017, Annex A

Table 4 contains all requirements related to the crypto suite state transitions.

Table 4 — Crypto suite requirements of ISO/IEC 29167-10:2017, Annex A

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5000	Annex A	Any combination of Start States and Transitions not listed in Table A.1 shall result in an error condition and consequently a transition to the **Initial** state.	M	Tag	By design
5010	Annex A	All other errors resulting from the execution of commands shall result in an error and consequently a transition to the **Initial** state.	M	Tag	By design

[a] All references are to ISO/IEC 29167-10, Annex A

[b] M: mandatory; items marked with "M" are mandatory and shall be tested for all devices.

O: optional; items marked with "O" are optional and shall be tested only for devices that support the feature that is indicated by the requirement.

[c] This column may define test patterns that are used for verification by demonstration.

6.3.5 Crypto suite requirements of ISO/IEC 29167-10:2017, Annex E

6.3.5.1 General

This subclause contains all requirements for the Protocol specific information.

6.3.5.2 Command definitions for ISO/IEC 29167-10:2017, E.1

Table 5 contains all requirements related to the concept of exchanging Messages and Responses.

Table 5 — Crypto suite requirements of ISO/IEC 29167-10:2017, E.1

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5100	E.1.1	For the implementation of this crypto suite, an air interface protocol shall support security commands that allow the exchange of data between the Interrogator and the Tag that has this crypto suite implemented.	M	Interrogator/ Tag	By design
5110	E.1.1	According to ISO/IEC 29167-1, the CSI for this crypto suite shall be defined as the 6-bit value 000000_2.	M	Interrogator/ Tag	By design
5120	E.1.2	A crypto suite shall identify for each security service and method in Table E.1 if it is mandatory, optional or prohibited.	M	Interrogator/ Tag	By design

[a] All references are to ISO/IEC 29167-10, E.1

[b] M: mandatory; items marked with “M” are mandatory and shall be tested for all devices.

O: optional; items marked with “O” are optional and shall be tested only for devices that support the feature that is indicated by the requirement.

[c] This column may define test patterns that are used for verification by demonstration.

6.3.5.3 Command definitions for ISO/IEC 29167-10:2017, E.2

This subclause defines the requirements for ISO/IEC 18000-3 MODE 1.

Table 6 contains all requirements related to the concept of exchanging Messages and Responses.

Table 6 — Crypto suite requirements of ISO/IEC 29167-10:2017, E.2 - ISO/IEC 18000-3 MODE 1

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5200	E.2.2	A crypto suite supporting ISO/IEC 18000-3, Mode 1 shall fulfil the protocol security command requirements as defined in this subclause	M	Tag	By design
5210	E.2.2	a) In accordance with the air interface standard, the Tag shall use the In-Process reply if the maximum execution time for an *Authenticate* command exceeds t1, as defined for the immediate reply.	M	Tag	By design
5220	E.2.2	b) The Tag shall ignore commands from an Interrogator during execution of a cryptographic operation.	M	Tag	By design
5230	E.2.2	c) The Tag shall support sending the contents of the ResponseBuffer in the reply to a ReadBuffer command if a ResponseBuffer is supported by the Tag.	M	Tag	By demonstration using test pattern 04
5240	E.2.2	d) The Tag may support a security timeout following a crypto error. The length of the security timeout shall be <200 ms.	M	Tag	By design
5250	E.2.2	e) The *Authenticate* command shall be supported for all supported authentication methods.	M	Tag	By design

Table 6 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5260	E.2.2	g) A Tag in any cryptographic state shall ignore an invalid command and stay in the current state. (Invalid commands means crypto commands with non-matching UID or CRC error.)	M	Tag	By design
5270	E.2.2	h) the Tag shall transition to the **Ready** state;	M	Tag	By design
5280	E.2.2	h) the Tag shall send an error code in case of a transition to the **Ready** state;	M	Tag	By design
5290	E.2.2	h) the Tag shall behave according to the error handling defined in ISO 18000-3, Mode 1;	M	Tag	By design
5300	E.2.2	h) if the Tag is in **Selected Secure** state, it shall transition to the **Ready** state.	M	Tag	By design
5310	E.2.2	i) The Tag shall remain in its current state after a Tag Authentication.	M	Tag	By design
5320	E.2.2	i) The Tag shall transition to **Selected Secure** state (corresponding to the **IA-OK** state) after processing successfully an Interrogator or Mutual Authentication.	M	Tag	By design
5330	E.2.3	In ISO/IEC 18000-3, Mode 1, the message to execute Tag authentication shall be transmitted to the Tag with the *Authenticate* or the *Challenge* command.	M	Tag	By demonstration using test pattern 03 and test pattern 05
5340	E.2.3	The message to execute Interrogator Authentication or Mutual Authentication shall be transmitted to the Tag with the *Authenticate* command.	M	Tag	By demonstration using test pattern 11
5350	E.2.3	The air interface shall return the response;	M	Tag	By design
5360	E.2.3	it shall be backscattered immediately after the command	M	Tag	By demonstration using test pattern 03
5370	E.2.3	and/or it shall be stored in the ResponseBuffer,	M	Tag	By demonstration and test pattern 05
5380	E.2.3	from where it shall be returned to the Interrogator with the *ReadBuffer* command.	M	Tag	By demonstration using test pattern 04
5390	E.2.3	For implementation of this document in ISO/IEC 18000-3, Mode 1, the CSI shall be expanded to the 8-bit value 00_h.	M	Tag	By design
5400	E.2.4	The crypto suite shall return the error conditions for the error handling described in the base standard.	M	Tag	By design
5410	E.2.5	Every payload parameter shall be transmitted LSBit and LSByte first.	M	Tag	By design
5420	E.2.5	The order of the payload parameters shall be transmitted in the sequence as defined in this document.	M	Tag	By design
5430	E.2.5	Bit field parameters shall be concatenated to achieve integer multiples of 8 bits.	M	Tag	By design

Table 6 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5440	E.2.7	Table E.3 shall contain zero or more pointers to an area with custom data within the Tag's memory. The maximum number of pointers is 16.	M	Tag	By design
5450	E.2.7	The chip manufacturer shall define which modes a particular Tag model supports for which memory profiles.	M		By design

[a] All references are to ISO/IEC 29167-10, E.2

[b] M: mandatory; items marked with "M" are mandatory and shall be tested for all devices.

O: optional; items marked with "O" are optional and shall be tested only for devices that support the feature that is indicated by the requirement.

[c] This column may define test patterns that are used for verification by demonstration.

6.3.5.4 Command definitions for ISO/IEC 29167-10:2017, E.3

This subclause is reserved to define the requirements for ISO/IEC 18000-3 MODE 3.

6.3.5.5 Command definitions for ISO/IEC 29167-10:2017, E.4

This subclause defines the requirements for ISO/IEC 18000-63.

Table 7 contains all requirements of ISO/IEC 18000-63.

Table 7 — Crypto suite requirements of ISO/IEC 29167-10:2017, E.4 - ISO/IEC 18000-63

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5600	E.4.1	A crypto suite supporting ISO/IEC 18000-63 shall fulfil the protocol security command requirements as defined in this subclause.	M		By design
5610	E.4.1	Optional choices shall be accepted for one-to-one communication. Reason: Since the Tag is singulated and the TID is known, supported options can be derived from it.	M		By design
5620	E.4.1	a) The Tag shall use the In-Process reply if the maximum execution time for an *Authenticate* command exceeds 20 ms.	M	Tag	By design
5630	E.4.1	b) The Tag shall ignore commands from an Interrogator during execution of a cryptographic operation.	M	Tag	By design
5640	E.4.1	c) The Tag may support sending the contents of the ResponseBuffer in the reply to an ACK command.	O	Tag	By design
5650	E.4.1	d) The Tag shall support sending the contents of the ResponseBuffer in the reply to a *ReadBuffer* command.	M	Tag	By demonstration using test pattern 04
5660	E.4.1	e) The Tag may support a security timeout following a crypto error. The length of the security timeout shall be <200 ms.	O	Tag	By design
5670	E.4.1	f) The *Authenticate* command shall be supported for all supported authentication methods.	M	Tag	By design

Table 7 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]
5680	E.4.1	g) The *Challenge* command may be supported for parts or all supported authentication methods.	O	Tag	By design
5690	E.4.1	h) A Tag in any cryptographic state other than the **initial** state (i.e. state after power-up) shall reset its cryptographic engine and transition to the open state upon receiving an invalid command.	M	Tag	By design
5700	E.4.1	i) For each Error Condition defined in the Crypto Suite: — The Tag shall transition to the **arbitrate** state. — The Tag shall send an Error Code in case of a transition to the **arbitrate** state.	M	Tag	By design
5710	E.4.1	j) The Tag shall remain in its current state after a Tag Authentication.	M	Tag	By design
5710	E.4.1	j) The Tag shall transition to the **secured** state after processing successfully an interrogator or mutual authentication.	M	Tag	By design
5720	E.4.2	In ISO/IEC 18000-63, the message to execute any authentication shall be transmitted to the Tag with the *Authenticate* or the *Challenge* command. The air interface shall return the response, either it shall be backscattered immediately after the command or it shall be stored in the ResponseBuffer, from where it shall be returned to the Interrogator with the *ReadBuffer* command.	M	Tag	By demonstration using test pattern 03, (Test pattern 7) test pattern 04 and test pattern 05
5730	E.4.2	ISO/IEC 18000-63 specifies an 8-bit CSI. For implementation of this document in ISO/IEC 18000-63, the CSI shall be expanded to the 8-bit value 00_h.	M	Tag	By design
5740	E.4.3	The error conditions of the crypto suite shall be returned to the Interrogator as error codes for the air interface.	M	Tag	By design
5750	E.4.4	ISO/IEC 18000-63 requires the definition of key properties. If an implementation does provide key properties for a key belonging to this crypto suite, it shall set the key properties to 0000_b.	M	Tag	By design
5760	E.4.5	Table E.5 shall contain zero or more pointers to an area with custom data within the Tag's memory.	M	Tag	By design
5770	E.4.5	The chip manufacturer shall define which modes a particular Tag model supports for which memory profiles.	M		By design

Table 7 *(continued)*

Item	Protocol subclause[a]	Requirement[a]	M/O[b]	Applies to	How verified[c]

[a] All references are to ISO/IEC 29167-10, E.4

[b] M: mandatory; items marked with "M" are mandatory and shall be tested for all devices.

O: optional; items marked with "O" are optional and shall be tested only for devices that support the feature that is indicated by the requirement.

[c] This column may define test patterns that are used for verification by demonstration.

7 Test patterns

7.1 General

This clause defines the test patterns for ISO/IEC 18000-3 MODE 1 and ISO/IEC 18000-63, both in combination with the ISO/IEC 29167-10:2017 AES crypto suite.

Said documents also contain the descriptions of the terms used in the test patterns.

7.2 Test pattern information

7.2.1 General

This subclause contains information that is necessary for understanding the description of the test patterns.

The test patterns use the "Authenticate" and the "ReadBuffer" commands.

The "Authenticate" command contains the "Message" parameter, the content of which is specified in ISO/IEC 29167-10 and defined for each applicable test pattern as Message=(content1, content 2, etc.), with the content that is applicable for the test pattern.

The test patterns use the term "Tag", also for ISO/IEC 18000-3 where the term "Tag" is referred to as VICC (vicinity integrated circuit card).

The test patterns use the terms **MAX_KeyID** for the number of keys and **MAX_Profiles** for the number of memory profiles that the Tag supports.

7.2.2 Information related to ISO/IEC 18000-3 MODE 1

7.2.2.1 General

The test patterns for ISO/IEC 18000-3 MODE 1 are derived from ISO/IEC 15693-3 and ISO/IEC 29167-10.

7.2.2.2 Clarification for the use of commands

The test patterns for ISO/IEC 15693 use the following commands:

```
•   Authenticate, with the request format
SOF;Flags[7:0];0x35;CSI=[7:0];Message;CRC16;EOF
```

(0x35 is the Command code for the *Authenticate* command.)

```
•   Challenge, with the request format
SOF;Flags[7:0];0x39;CSI=[7:0];Message;CRC16;EOF
```

(0x39 is the Command code for the *Authenticate* command.)

```
•   ReadBuffer, with the request format:
SOF;Flags[7:0];0x3A;CRC16;EOF
```

(0x3A is the Command code for the ReadBuffer command.)

7.2.2.3 Error code handling

All the test patterns verifying an error response shall be executed in addressed or selected mode.

An error response may contain a specific error code or the code indicating “Other error” (ISO/IEC 15693-3:2019, Table 7 — Response error code: definition Error with no information given or a specific error code is not supported).

Instead of transmitting an error response ISO/IEC 18000-3 MODE 1 also allows to not send a response at all (no reply to the command).

7.2.3 Information related to ISO/IEC 18000-63

7.2.3.1 General

The test patterns are derived from ISO/IEC 18000-63 and ISO/IEC 29167-10.

7.2.3.2 Clarification for the use of the test patterns

The test patterns shall use the value for CSI[7:0] that is used on the Tag for the implementation of ISO/IEC 29167-10 (usually this value will be 0x00).

In the description of the test patterns for ISO/IEC 18000-63,

— Miller2 stands for “Miller Subcarrier Sequence M=2”

— Miller4 stands for “Miller Subcarrier Sequence M=4”

7.2.3.3 Error code handling

ISO/IEC 18000-63:2015, 6.3.2.12.3.10 (*Authenticate*) states:

If a Tag receives an *Authenticate* specifying an unsupported CSI, an improperly formatted or not-executable message, or an improper cryptographic parameter, then the Tag shall not execute the *Authenticate* and instead treat the command’s parameters as unsupported (return a "Not Supported Error" response).

ISO/IEC 18000-63:2015, Table I.2 (Tag error codes) also allows the return of the "Non-specific error" code (The Tag does not support error-specific codes).

7.3 Test pattern descriptions

7.3.1 General

This subclause defines the details of all test pattern parameters for ISO/IEC 18000-3 MODE 1 and ISO/IEC 18000-63.

7.3.2 Test pattern 01 (TAM reject message when "AuthMethod" is '11')

7.3.2.1 General

This test pattern verifies if the *Authenticate* command is rejected when "AuthMethod" is '11'.

For the execution of this test pattern KeyID[7:0] shall contain a value that the Tag can use for TAM1.

7.3.2.2 Test pattern for ISO/IEC 18000-3 MODE 1

```
Authenticate
Message=(AuthMethod=11,CustomData=0,TAM1_RFU=00000,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)
```

The test pattern passes when the Tag returns a “Not Supported” error message.

7.3.2.3 Test pattern for ISO/IEC 18000-63

```
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=11,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3))
```

The test pattern passes when the Tag returns a “Not Supported” error message.

7.3.3 Test pattern 02 (TAM1 execution and error handling)

7.3.3.1 General

This test pattern verifies if the Tag properly executes the error handling if the length of the TAM1 message is <> 96 bits, if TAM1_RFU <> “00000” or if the message is rejected when the KeyID is not supported by the Tag.

For the execution of this test pattern KeyID[7:0] shall contain a value that the Tag can use for TAM1.

7.3.3.2 Test pattern for ISO/IEC 18000-3 MODE 1

```
// Step 1: TAM1 (with 96-bits message length, 80-bits IChallenge_TAM1)
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)

// Step 2: TAM1 (with 88-bits message length, 72-bits IChallenge_TAM1)
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965C)

// Step 3: TAM1 (with 104-bits message length, 88-bits IChallenge_TAM1)
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CCFF)

// Step 4: TAM1 (with TAM1_RFU=00001)
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00001,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)

// Step 5: TAM1 (with KeyID=MAX_KeyID+1)
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID=MAX_KeyID+1,IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)
```

The test pattern passes when all the following occurred:

1. Content of the decrypted response field for the first TAM1 is:

   ```
   C_TAM1=0x96C5;TRnd_TAM1[31:0];IChallenge_TAM1=0xD53600FAA9B4C1965CC3
   ```

2. Response to the second TAM1 is “Other Error” or no reply
3. Response to the third TAM1 is “Other Error” or no reply
4. Response to the fourth TAM1 is “Not Supported Error” or “Other error” or no reply
5. Response to the fifth TAM1 is “Not Supported Error” or “Other error” or no reply

7.3.3.3 Test pattern for ISO/IEC 18000-63, 6.3.1.6.4

```
// Step 1: TAM1 (with 96-bits message length, 80-bits IChallenge_TAM1)
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3))

// Step 2: TAM1 (with 88-bits message length, 72-bits IChallenge_TAM1)
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965C))

// Step 3: TAM1 (with 104-bits message length, 88-bits IChallenge_TAM1)
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3FF)

// Step 4: TAM1 (with TAM1_RFU=00001)
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00001,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3))

// Step 5: TAM1 (with KeyID=MAX_KeyID+1)
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID=MAX_KeyID+1,IChallenge_TAM1=0xD53600FAA9B4C1965CC3))
```

The test pattern passes when all the following has occurred:

1. Content of the decrypted response field for the first TAM1 is:

   ```
   C_TAM1=0x96C5;TRnd_TAM1[31:0];IChallenge_TAM1=0xD53600FAA9B4C1965CC3
   ```

2. Response to the second TAM1 is “Other Error” or "Not Supported Error" or "Non-specific error"
3. Response to the third TAM1 is “Other Error” or "Not Supported Error" or "Non-specific error"
4. Response to the fourth TAM1 is “Not Supported Error” or "Non-specific error"
5. Response to the fifth TAM1 is “Not Supported Error” or "Non-specific error"

7.3.4 Test pattern 03 (TAM1 execution for all keys)

7.3.4.1 General

This test pattern verifies if AuthMethod '00' (TAM1) works properly for all keys that the Tag supports for TAM1.

This pattern shall be executed for all values of KeyID[7:0] that the Tag can use for TAM1.

7.3.4.2 Test pattern for ISO/IEC 18000-3 MODE 1

```
// TAM1
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID=[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)
```

The test pattern passes when the content of the decrypted response field is:

```
C_TAM1=0x96C5;TRnd_TAM1[31:0];IChallenge_TAM1=0xD53600FAA9B4C1965CC3
```

7.3.4.3 Test pattern for ISO/IEC 18000-63, 6.3.1.6

```
// TAM1
Query (Tari=6,25µs; BLF=640; FM0, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=1;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,
KeyID=[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3))
```

The test pattern passes when the content of the decrypted response field is:

```
C_TAM1=0x96C5;TRnd_TAM1[31:0];IChallenge_TAM1=0xD53600FAA9B4C1965CC3
```

7.3.5 Test pattern 04 (TAM1 store Tag reply in the response buffer)

7.3.5.1 General

This test pattern verifies if the Tag properly stores the Tag reply in the response buffer.

This test pattern shall be executed for a KeyID that the Tag can use for TAM1.

7.3.5.2 Test pattern for ISO/IEC 18000-3 MODE 1

This test pattern is only applicable for Tags that have implemented the response buffer.

Execution of the test pattern can start if the ResponseBuffer Validity_flag (b2) of the Response flags is set to zero.

```
// Step 1: Verify that ReadBuffer command is not accepted when the ResponseBuffer
Validity_flag (b2) of the Response flags is set to zero and an error response occurs
ReadBuffer

// Step 2: TAM1
Authenticate
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)

// Verify that the ResponseBuffer Validity_flag =1

// Step 3: Read Tag reply from response buffer
ReadBuffer

// Decrypt and verify response
```

The test pattern passes when all the following has occurred:

1. First ReadBuffer is not accepted and an error code is returned.
2. Verify that the ResponseBuffer Validity_flag =1 after TAM1.
3. The decrypted data for the first TAM1, recovered by ReadBuffer, is:

```
C_TAM1=0x96C5;TRnd_TAM1;IChallenge_TAM1=0xD53600FAA9B4C1965CC3.
```

7.3.5.3 Test pattern for ISO/IEC 18000-63, 6.3.1.6.4

```
// Step 1: Read the XPC_W1 word and verify C-flag=0
Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Read (MemBank=01;WordPtr=0x21;WordCnt=0x01)

// Step 2: Verify that ReadBuffer command is not accepted when C-flag=0 and an error
response occurs
ReadBuffer
(RFU=0;WordPtr=0x000;BitCount=0x000)

// Step 3: First TAM1
Authenticate
(SenRep=0;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3))

// Step 4: Read the XPC_W1 word and verify C-flag=1
Read
(MemBank=01;WordPtr=0x21;WordCnt=0x01)

// Read Tag reply from response buffer
ReadBuffer
(RFU=0;WordPtr=0x000;BitCount=0x000)

// Decrypt and verify response

// Step 5: Second TAM1
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0x96564402375796C69664))
```

The test pattern passes when all the following has occurred:

1. C-flag=0 for first read of XPC_W1 word.
2. First ReadBuffer is not accepted and an error code is returned.
3. No response field is included in the tag reply to the first Authenticate.
4. C-flag=1 for second read of XPC_W1 word. The decrypted data for the first TAM1 recovered by ReadBuffer is:

```
C_TAM1=0x96C5;TRnd_TAM1;IChallenge_TAM1=0xD53600FAA9B4C1965CC3.
```

5. The decrypted data for the second TAM1 response field is:

```
C_TAM1=0x96C5;TRnd_TAM1;IChallenge_TAM1=0x96564402375796C69664.
```

7.3.6 Test pattern 05 (TAM1 with Challenge, read Tag reply from the response buffer)

7.3.6.1 General

This test pattern verifies if the Tag properly stores the Tag reply in the response buffer after a Challenge command.

This test pattern is only applicable for Tags that have implemented the Challenge command.

This test pattern shall be executed for a KeyID that the Tag can use for TAM1.

7.3.6.2 Test pattern for ISO/IEC 18000-3 MODE 1

```
// Step 1: Use Challenge command in Unaddressed mode, using TAM1
Challenge
Message=(AuthMethod=00,CustomData=0,TAM1_RFU=00000,KeyID[7:0],IChallenge_
TAM1=0xD53600FAA9B4C1965CC3)

// Wait sufficient time for the Tag to finalize the cryptographic calculation

// Step 2: Read reply from response buffer
ReadBuffer

// Decrypt and verify response
```

The test pattern passes when all the following has occurred:

1. The Tag shall not reply to the Challenge command.

2. The decrypted data for TAM1, recovered by ReadBuffer, is:

```
C_TAM1=0x96C5;TRnd_TAM1;IChallenge_TAM1=0xD53600FAA9B4C1965CC3.
```

7.3.6.3 Test pattern for ISO/IEC 18000-63, 6.3.1.6.4

```
// Step 1: Challenge command using TAM1
Challenge
(IncRepLen=0;Immed=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=0,TAM1_
RFU=00000,KeyID[7:0],IChallenge_TAM1=0xD53600FAA9B4C1965CC3))

// Wait sufficient time for the Tag to finalize the cryptographic calculation

// Step 2: Read reply from response buffer
Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
ReadBuffer
(RFU=0;WordPtr=0x000;BitCount=0x000)

// Decrypt and verify response
```

The test pattern passes when all the following occurred:

1. The Tag shall not reply to the Challenge command.

2. The decrypted data for TAM1, recovered by ReadBuffer, is:

```
C_TAM1=0x96C5;TRnd_TAM1;IChallenge_TAM1=0xD53600FAA9B4C1965CC3
```

7.3.7 Test pattern 06 (TAM2 execution and error handling)

7.3.7.1 General

This test pattern verifies if the Tag properly executes the error handling if the length of the TAM2 message is <> 120 bits, if TAM2_RFU <> "000", if it does not support key[KeyID].*ENC_key*, if the memory profile specified in Profile is not supported or if the block of custom data specified by BlockSize, Profile, Offset and BlockCount is not supported.

For the execution of this test pattern KeyID[7:0] shall contain a value that the Tag can use for TAM2.

7.3.7.2 Test pattern for ISO/IEC 18000-3 MODE 1 for TAM2_Rev=0

TAM2_Rev=0 was defined in ISO/IEC 29167-10 and is not applicable for ISO/IEC 18000-3 MODE 1 since there are no HF products on the market that comply with that version of the standard.

7.3.7.3 Test pattern for ISO/IEC 18000-3 MODE 1 for TAM2_Rev=1

```
// Step 1: TAM2 (with 120-bits message length, 80-bits IChallenge_TAM2)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_
RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Offset=0x000,BlockC
ount=0001,ProtMode=0001)

// Step 2: TAM2 (with 112-bits message length, 72-bits IChallenge_TAM2)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_
RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965C,Profile[3:0],Offset=0x000,BlockCou
nt=0001,ProtMode=0001)

// Step 3: TAM2 (with 128-bits message length, 88-bits IChallenge_TAM2)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_
RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3FF,Profile[3:0],Offset=0x000,Blo
ckCount=0001,ProtMode=0001)

// Step 4: TAM2 (with TAM2_RFU=001)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_
RFU=001,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Offset=0x000,BlockC
ount=0001,ProtMode=0001)

// Step 5: TAM2 (with KeyID=MAX_KeyID+1)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_RFU=000,KeyID=MAX_
KeyID+1,IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Offset=0x000,BlockCount=0001,P
rotMode=0001)

// Step 6: TAM2 (with Profile=MAX_Profile+1)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_
RFU=000,KeyID=[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile=MAX_Profile+1,Offset=0x
000,BlockCount=0001,ProtMode=0001)

// Step 7: TAM2 (with block of custom data, specified by BlockSize, Profile, Offset and
BlockCount, that is not supported by the Tag)
Authenticate
Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,TAM2_REV=1,TAM2_
RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Offset[11:0],BlockC
ount[3:0],ProtMode=0001)
```

The test pattern passes when all the following occurred:

1. Content of the decrypted response field for the first TAM2 is:

```
C_TAM2=0x96C5;TRnd_TAM2;IChallenge_TAM2=0xD53600FAA9B4C1965CC3;CUSTOMDATA
```

2. Response to the second TAM2 is “Other Error” or no reply
3. Response to the third TAM2 is “Other Error” or no reply
4. Response to the fourth TAM2 is “Not Supported Error” or “Other Error” or no reply
5. Response to the fifth TAM2 is “Not Supported Error” or “Other Error” or no reply
6. Response to the sixth TAM2 is “Not Supported Error”or “Other Error” or no reply
7. Response to the seventh TAM2 is “Memory Overrun Error” or “Other Error” or no reply

7.3.7.4 Test pattern for ISO/IEC 18000-63:2015 for TAM2_Rev=0

```
// Step 1: TAM2 (with 96-bits IChallenge_TAM2)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
```

```
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Off
set=0x000,BlockCount=0001,ProtMode=0001))

// Step 2: TAM2 (with 88-bits IChallenge_TAM2)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965C,Profile[3:0],Offse
t=0x000,BlockCount=0001,ProtMode=0001))

// Step 3: TAM2 (with 104-bits IChallenge_TAM2)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3FF,Profile[3:0],O
ffset=0x000,BlockCount=0001,ProtMode=0001))

// Step 4: TAM2 (with TAM2_RFU=001)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=001,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Off
set=0x000,BlockCount=0001,ProtMode=0001))

// Step 5: TAM2 (with KeyID=MAX_KeyID+1)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=000,KeyID=MAX_KeyID+1,IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3
:0],Offset=0x000,BlockCount=0001,ProtMode=0001))

// Step 6: TAM2 (with Profile=MAX_Profile+1)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=000,KeyID=MAX_KeyID+1,IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile=
MAX_Profile+1,Offset=0x000,BlockCount=0001,ProtMode=0001))

// Step 7: TAM2 (with block of custom data, specified by BlockSize, Profile, Offset and
BlockCount, that is not supported by the Tag)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=0,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Off
set[11:0],BlockCount[3:0],ProtMode=0001))
```

The test pattern passes when all the following occurred:

1. Content of the decrypted response field for the first TAM2 is:

`C_TAM2=0x96C5;TRnd_TAM2;IChallenge_TAM2=0xD53600FAA9B4C1965CC3;CUSTOMDATA`

2. Response to the second TAM2 is “Other Error” or "Not Supported Error" or "Non-specific error"

3. Response to the third TAM2 is “Other Error” or "Not Supported Error" or "Non-specific error"

4. Response to the fourth TAM2 is “Not Supported Error” or "Non-specific error"

5. Response to the fifth TAM2 is “Not Supported Error” or "Non-specific error"

6. Response to the sixth TAM2 is “Not Supported Error” or "Non-specific error"

7. Response to the seventh TAM2 is “Memory Overrun Error” or "Not Supported Error" or "Non-specific error"

7.3.7.5 Test pattern for ISO/IEC 18000-63:2015 for TAM2_Rev=1

```
// Step 1: TAM2 (with 96-bits IChallenge_TAM2)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=1,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Off
set=0x000,BlockCount=0001,ProtMode=0001))

// Step 2: TAM2 (with 88-bits IChallenge_TAM2)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=1,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965C,Profile[3:0],Offse
t=0x000,BlockCount=0001,ProtMode=0001))

// Step 3: TAM2 (with 104-bits IChallenge_TAM2)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=1,TAM2_RFU=000,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3FF,Profile[3:0],O
ffset=0x000,BlockCount=0001,ProtMode=0001))

// Step 4: TAM2 (with TAM2_RFU=001)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=1,TAM2_RFU=001,KeyID[7:0],IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3:0],Off
set=0x000,BlockCount=0001,ProtMode=0001))

// Step 5: TAM2 (with KeyID=MAX_KeyID+1)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=1,TAM2_RFU=000,KeyID=MAX_KeyID+1,IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile[3
:0],Offset=0x000,BlockCount=0001,ProtMode=0001))

// Step 6: TAM2 (with Profile=MAX_Profile+1)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
Req_RN
Authenticate
(SenRep=1;IncRepLen=0;CSI=[7:0];Length;Message=(AuthMethod=00,CustomData=1,BLOCKSIZE=0,
TAM2_REV=1,TAM2_RFU=000,KeyID=MAX_KeyID+1,IChallenge_TAM2=0xD53600FAA9B4C1965CC3,Profile=
MAX_Profile+1,Offset=0x000,BlockCount=0001,ProtMode=0001))

// Step 7: TAM2 (with block of custom data, specified by BlockSize, Profile, Offset and
BlockCount, that is not supported by the Tag)
Query (Tari=12,5µs; BLF=320; Miller2, S0)
ACK
```