INTERNATIONAL STANDARD

ISO/IEC 19823-13

First edition
2018-04

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

Part 13: Cryptographic Suite Grain-128A

Technologies de l'information — Conformance test methods for security service crypto suites —

Partie 13: Suite cryptographique Grain-128A

Reference number
ISO/IEC 19823-13:2018(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

4 Test methods 2

4.1 General 2

4.2 By demonstration 2

4.3 By design 2

5 Test methods in respect to the ISO/IEC 18000 parts 2

5.1 Test requirements for ISO/IEC 18000-62 interrogators and tags 2

6 Test methods in respect to the ISO/IEC 29167-13 interrogators and tags 3

6.1 Test map for optional features 3

6.2 Crypto suite requirements 3

6.3 Test patterns 14

Bibliography 21

Foreword

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

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

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

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

For an explanation on the 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 the following URL: 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*.

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

Introduction

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

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

These relations mean that for a product that is claimed to be compliant to a pair of ISO/IEC 18000-n and ISO/IEC 29167-m then the test methods of ISO/IEC 18047-n and ISO/IEC 19823-m 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 Grain-128A crypto suite as standardized in ISO/IEC 29167-13.

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

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

Part 13: Cryptographic Suite Grain-128A

1 Scope

This document describes test methods for determining the conformance of security crypto suites with the specifications given in ISO/IEC 29167-13.

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

The conformance parameters are the following:

- parameters that apply directly affecting system functionality and inter-operability;
- protocol including commands and replies; and
- nominal values and tolerances.

Unless otherwise specified, the tests in this document are applied exclusively to RFID tags and interrogators defined in the ISO/IEC 18000 series using ISO/IEC 29167-13.

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 19762 (all parts), *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary*

ISO/IEC 18000-62, *Information technology — Radio frequency identification for item management — Part 62: Parameters for air interface communications at 860 MHz to 960 MHz Type B*

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 29167-13:2015, *Information technology — Automatic identification and data capture techniques — Part 13: Crypto suite Grain-128A security services for air interface communications*

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

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 (all parts) and ISO/IEC 29167-13 apply.

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

- IEC Electropedia: available at https://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

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 29167-13. As the parts of ISO/IEC 19823 are always tested in relation with ISO/IEC 18047 a duplication of information requirements, and specifications should be avoided.

Clause 5 defines elements that are assumed to be covered in the respective ISO/IEC 18047 parts and therefore shall not be addressed in an ISO/IEC 19823 part. Only if ISO/IEC 18047 does not define them, then they may be defined in ISO/IEC 19823, although a revision of ISO/IEC 18047 should be the preferred option.

Clause 6 defines elements that are not expected to be covered by ISO/IEC 18047 and these shall be addressed in the respective ISO/IEC 19823 part.

4.2 By demonstration

Laboratory testing of one, or (if required for statistical reasons), multiple products, processes, or services to ensure compliance. A test laboratory that meets ISO/IEC 17025 shall perform 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

Design parameters and/or theoretical analysis that ensure compliance. A vendor submitting a component or system for compliance 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 in respect to the ISO/IEC 18000 parts

5.1 Test requirements for ISO/IEC 18000-62 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 Setup of test equipment.

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

— Clause 7 Conformance tests for ISO/IEC 18000-62.

6 Test methods in respect to the ISO/IEC 29167-13 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 template to report the test results. Furthermore, it is used to refer to the test requirements in 6.2.

Table 1 — Test map for optional features

#	Feature	Additional requirement	Mark items to be tested for supplied product	Test results
1	TA	Shall be tested with the authenticate command of the declared ISO/IEC 18000 part		
2	IA	Shall be tested with the authenticate command of the declared ISO/IEC 18000 part		
3	Secure Authenticated Communication	Shall be tested with the SecureComm command of the declared ISO/IEC 18000 part		
4	Key update	Shall be tested with the SecureComm command of the declared ISO/IEC 18000 part		
2	Number of keys supported			

Table 2 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 Crypto suite requirements

This clause contains all requirements of ISO/IEC 29167-13.

6.2.1 Crypto suite requirements of ISO/IEC 29167-13:2015 in Clauses 1 to 8 and Annexes A to C

All the requirements of ISO/IEC 29167-13:2015 in Clauses 1 to 8 and Annexes A to C shall apply, inherently by design only.

6.2.2 Crypto suite requirements of ISO/IEC 29167-13:2015 in Clauses 9 to 12 and Annex E

Table 2 contains all requirements of ISO/IEC 29167-13:2015 in Clauses 9 to 12 and Annex E.

The column MO (Mandatory/optional) has the following content:

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.

Table 2 — Crypto suite requirements

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
1	9	The Tag's air interface protocol logic shall provide an external reset to the Tag crypto engine which shall set **INIT**=FALSE, **TA**=FALSE, **IA**=FALSE and ERROR=FALSE before transition to the **CS-Reset** state.	M	Tag	By design
2	9	The **CS-Reset** state shall process crypto commands from the Tag's air interface protocol logic only when **ERROR**=FALSE. If an error condition exists then the Tag crypto engine shall set ERROR=TRUE and remain in the **CS-Reset** state.	M	Tag	By design
3	9	If an error condition exists then the Tag crypto engine shall set ERROR=TRUE and remain in the **CS-Reset** state.	M	Tag	By design
4	9	The Tag shall report an error condition if it receives a CryptoCommCmd, CryptoSecCommCmd or CryptoKeyUpdate command in the **CS-Reset** state.	M	Tag	By design
5	9	The Tag shall check a CryptoAuthCmd payload for any error conditions.	M	Tag	By design
6	9	The Tag shall report an error condition if Step ≠ 00_b in the **CS-Reset** state.	M	Tag	By demonstration using Test Pattern 3
7	9	The Tag shall report an error condition if the KeyID value is not supported by the Tag.	M	Tag	By demonstration using Test Pattern 2 (only if TA is supported), Test Pattern 10 (only if IA is supported) and Test Pattern 16
8	9	The Tag shall report an error condition if AuthMethod=00_b and the Tag does not support Tag authentication.	M	Tag	By design
9	9	The Tag shall report an error condition if AuthMethod=00_b and the Options selected are not supported by the Tag CSFeatures.	O	Tag	By design
10	9	The Tag shall report an error condition if AuthMethod=01_b and the Tag does not support Interrogator authentication.	M	Tag	By design
11	9	The Tag shall report an error condition if AuthMethod=01_b and Options ≠ 0000_b.	O	Tag	By demonstration using Test Pattern 9
12	9	The Tag shall report an error condition if AuthMethod=10_b and Options ≠ 0000_b.	M	Tag	By demonstration using Test Pattern 15
13	9	The Tag shall report an error condition if AuthMethod=11_b and the Tag does not support a vendor defined authentication.	M	Tag	By design
14	9	If no error condition exists, the Tag shall transition to the **CS-Init** state.	M	Tag	By design
15	10.1	The authentication method to be performed shall be specified by the 2-bit value AuthMethod which is defined in Table 2.	M	Tag, Interrogator	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
16	10.1	If AuthMethod="00b" the Tag shall parse the Message for Tag Authentication as described in section 10.2	O	Tag	By demonstration using Test Pattern 1
17	10.1	If AuthMethod="01b" the Tag shall parse the Message Interrogator Authentication as described in section 10.3	O	Tag	By demonstration using Test Pattern 8
18	10.1	If AuthMethod="10b" the Tag shall parse the Message for Mutual Authentication as described in section 10.4	M	Tag	By demonstration using Test Pattern 14
19	10.1	Some of the authentication methods require multiple steps to be performed in a specific sequence. The current step in the sequence shall be specified by the 2-bit value Step as defined in Table 3.	M	Tag, Interrogator	By design
20	10.1	During step 0 of an authentication method, the Tag shall provide an 8-bit value CSFeatures which is used to indicate which of the optional Grain-128A CS features are supported by the Tag.	M	Tag	By design
21	10.1	During step 0 and 1 of an authentication method, the Interrogator shall provide a 4-bit value Options	M	Interrogator	By design
22	10.2.1	The Tag authentication method uses a challenge-response protocol having one pair of message exchange (see Figure 2).	O	Interrogator, Tag	By design
23	10.2.2	For Tag authentication the Interrogator shall generate a 48-bit random number for use as IRandomNumber and issue the challenge to the Tag with the TA.1 Payload as specified in Table 6.	O	Interrogator	By design
24	10.2.3	The Tag shall generate a 48-bit random number for use as TRandomNumber. The Tag crypto engine shall be initialized for Tag authentication using TRandomNumber, IRandomNumber and the crypto key specified by KeyID. The crypto engine then shall generate the Tag keystream.	O	Tag	By design
25	10.2.3	The Tag shall respond to the challenge from the Interrogator with the TA.1 Payload as specified in Table 7.	O	Tag	By design
26	10.2.3	The Tag shall transition to the **TA.1** state after the response to the Interrogator and shall set **TA**=TRUE.	O	Tag	By design
27	10.2.4	The Interrogator shall be initialized for Tag authentication using TRandomNumber, IRandomNumber and the crypto key specified by KeyID. The crypto engine shall then generate the Interrogator keystream.	O	Interrogator	By design
28	10.2.4	The Interrogator shall compare the Tag keystream with the Interrogator keystream to authenticate the Tag and accepts it as valid if they are equal.	O	Interrogator	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
29	10.3.1	The Interrogator authentication method uses a challenge-response protocol having two pairs of message exchange as shown in (see Figure 3).	O	Interrogator, Tag	By design
30	10.3.2	In the first step of the Interrogator authentication process, the Interrogator shall generate a 48-bit random number for use as IRandomNumber and request a challenge from the Tag using the IA.1 Payload, as specified in Table 8.	O	Interrogator	By design
31	10.3.3	The Tag shall generate a 48-bit random number for use as TRandomNumber. The Tag crypto engine shall be initialized for Interrogator authentication using TRandomNumber, IRandomNumber and the crypto key specified by KeyID.	O	Tag	By design
32	10.3.3	The Tag shall respond with the challenge to the Interrogator with the IA.1 Payload as specified in Table 9.	O	Tag	By design
33	10.3.3	The Tag shall transition to the **IA.1** state after the response to the Interrogator.	O	Tag	By design
34	10.3.4	In the second step, the Interrogator crypto engine shall be initialized for Interrogator authentication using TRandomNumber, IRandomNumber and the crypto key specified by KeyID. The crypto engine shall then generate the Interrogator keystream.	O	Interrogator	By design
35	10.3.4	The Interrogator shall respond to the challenge from the Tag with the IA.2 Payload as specified in Table 10.	O	Interrogator	By design
36	10.3.5	The Tag shall check the crypto command and payload for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **IA.1** state.	O	Tag	By design
37	10.3.5	The Tag shall report an error condition if it receives a CryptoCommCmd, CryptoSecCommCmd or CryptoKeyUpdate command in the **IA.1** state.	O	Tag	By design
38	10.3.5	The Tag shall report an error condition if AuthMethod ≠ 01_b in the IA.2 payload.	O	Tag	By design
39	10.3.5	The Tag shall report an error condition if Step ≠ 01_b in the IA.2 payload.	O	Tag	By design
40	10.3.5	The Tag shall report an error condition if the KeyID value is not the same as used for the IA.1 payload.	O	Tag	By design
41	10.3.5	The Tag shall report an error condition if the selected Options are not supported by the Tag CSFeatures.	O	Tag	By design
42	10.3.5	If no error condition exists, the Tag crypto engine shall generate the Tag keystream and compare it with the Interrogator keystream. It shall accept the Interrogator as valid if the parameters are equal.	O	Tag	By demonstration using Test Pattern 8 and Test Pattern 11

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
43	10.3.5	The Tag shall respond with the IA.2 Payload as specified in Table 11.	O	Tag	By design
44	10.3.5	If the Interrogator authentication succeeded, the Tag shall transition to the **IA.2** state after the response to the Interrogator and set **IA**=TRUE.	O	Tag	By design
45	10.3.5	If the Interrogator authentication failed, the Tag shall transition to the **IA.2** state after the response to the Interrogator and set **ERROR**=True.	O	Tag	By design
46	10.4.1	The mutual authentication method uses a challenge-response protocol having two pairs of message exchange (see Figure 4).	M	Interrogator, Tag	By design
47	10.4.2	In the first step of the mutual authentication process, the Interrogator shall generate a 48-bit random number for use as IRandomNumber and request a challenge from the Tag using the MA.1 Payload, as specified in Table 12.	M	Interrogator	By design
48	10.4.3	The Tag shall generate a 48-bit random number for use as TRandomNumber. The Tag crypto engine shall be initialized for mutual authentication using the crypto key specified in KeyID, TRandomNumber and IRandomNumber.	M	Tag	By design
49	10.4.3	The Tag shall respond with the challenge to the Interrogator with the MA.1 Payload as specified in Table 13.	M	Tag	By design
50	10.4.3	The Tag shall transition to the **MA.1** state after the response to the Interrogator.	M	Tag	By design
51	10.4.4	In the second step, the Interrogator shall be initialized for mutual authentication using TRandomNumber, IRandomNumber and the crypto key specified by KeyID. The crypto engine shall then generate the Interrogator keystream.	M	Interrogator	By design
52	10.4.4	The Interrogator shall respond to the challenge from the Tag with the MA.2 Payload as specified in Table 14.	M	Interrogator	By design
53	10.4.5	The Tag shall check the crypto command and payload for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **MA.1** state.	M	Tag	By design
54	10.4.5	The Tag shall report an error condition if it receives a CryptoCommCmd, CryptoSecCommCmd or CryptoKeyUpdate command in the **MA.1** state.	M	Tag	By design
55	10.4.5	The Tag shall report an error condition if AuthMethod ≠ 10_b in the MA.2 payload.	M	Tag	By design
56	10.4.5	The Tag shall report an error condition if Step ≠ 01_b in the MA.2 payload.	M	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
57	10.3.5	The Tag shall report an error condition if the KeyID value is not the same as used for the MA.1 payload.	M	Tag	By design
58	10.3.5	The Tag shall report an error condition if the selected Options are not supported by the Tag CSFeatures.	M	Tag	By design
59	10.4.5	If no error condition exists, the Tag crypto engine shall generate the Tag keystream and compare it with the Interrogator keystream. It shall accept the Interrogator as valid if the parameters are equal.	M	Tag	By demonstration using Test Pattern 14 and Test Pattern 17
60	10.4.5	If the Interrogator is invalid, the Tag shall transition to the **MA.2** state after the response to the Interrogator and set **ERROR**=True.	M	Tag	By design
61	10.4.5	If the Interrogator is valid, the Tag crypto engine shall generate a new value for the Tag keystream.	M	Tag	By design
62	10.4.5	The Tag shall transition to the **MA.2** state after the response to the Interrogator and set TA=**IA**=TRUE.	M	Tag	By design
63	10.4.5	The Tag shall respond with the MA.2 Payload as specified in Table 15.	M	Tag	By design
64	10.4.6	The Interrogator shall check the authentication status from the Tag and if it is OK, the Interrogator crypto engine shall generate a new value for the Interrogator keystream.	M	Interrogator	By design
65	10.4.6	The Interrogator shall use the updated keystreams to authenticate the Tag. The Tag is accepted as valid the Tag keystream and the Interrogator keystream are equal.	M	Interrogator	By design
66	11.1	Authentication integrity shall be maintained for an Interrogator authentication and a mutual authentication, it is optional to maintain the authentication integrity of a Tag authentication.	M	Interrogator, Tag	By design
67	11.1	Authentication integrity shall be performed using authenticated communication and/or secure authenticated communication.	M	Interrogator, Tag	By design
68	11.1	A Message Authentication Code (MAC) shall be used to provide the integrity protection.	M	Interrogator, Tag	By design
69	11.1	The Interrogator shall select between using a MAC32 or a MAC64 via the Options parameter during the authentication process.	M	Interrogator	By design
70	11.2	If a Tag is authenticated as a result of Tag authentication, the Interrogator may use authenticated communication.	O	Interrogator	By demonstration using Test Pattern 4
71	11.2	The TA.1 state shall process crypto responses from the Tag's air interface protocol logic only when **ERROR**=FALSE.	O	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
72	11.2	The Tag shall check the crypto responses for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **TA.1** state.	O	Tag	By design
73	11.2	An error condition shall occur for any CryptoSecCommResp or CryptoAuthResp in the **TA.1** state.	O	Tag	By design
74	11.2	If no error condition exists, the Tag shall provide integrity protection for the Tag response in the CryptoCommResp payload (see Table 17).	O	Tag	By design
75	11.2	Integrity of the Tag response shall be performed with the addition of an 8-bit value of 00_h followed by a MAC.	O	Tag	By design
76	11.2	The Tag shall remain in the **TA.1** state after the response is sent to the Interrogator.	O	Tag	By design
77	11.2	The Interrogator shall generate a MAC for the Tag response within the CryptoCommResp payload to authenticate the Tag response.	O	Interrogator	By design
78	11.2	The Interrogator shall accept the response as valid if the MAC from the Tag equals the MAC from the Interrogator.	O	Interrogator	By design
79	11.2	If an Interrogator is authenticated as a result of Interrogator authentication, then it shall maintain the integrity of the authentication during subsequent communications with the singulated Tag.	O	Interrogator	By demonstration using Test Pattern 12 and Test Pattern 13
80	11.2	The Interrogator shall provide integrity protection for the command in the CryptoCommCmd payload (see Table 16).	O	Interrogator	By design
81	11.2	Integrity of the Interrogator command shall be performed with the addition of an 8-bit value of 00_h followed by a MAC.	O	Interrogator	By design
82	11.2	The **IA.2** state shall process crypto commands from the Tag's air interface protocol logic only when **ERROR**=FALSE.	O	Tag	By design
83	11.2	The Tag shall check the crypto commands for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **IA.2** state.	O	Tag	By design
84	11.2	An error condition shall occur for any CryptoAuthCmd, CryptoKeyUpdate or CryptoSecCommCmd in the **IA.2** state.	O	Tag	By design
85	11.2	If no error condition exists, the Tag shall generate a MAC for the Interrogator command within the CryptoCommCmd payload to authenticate the Interrogator command.	O	Tag	By design
86	11.2	The Tag shall accept the Interrogator command as valid if the MAC from the Interrogator equals the MAC from the Tag.	O	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
87	11.2	If the Interrogator command is invalid, the Tag crypto engine shall set **ERROR**=TRUE and the Tag shall remain in the **IA.2** state.	O	Tag	By design
88	11.2	If a Tag and Interrogator are both authenticated as a result of mutual authentication, then both shall maintain the integrity of the authentication during subsequent communications with the singulated Tag.	M	Interrogator, Tag	By demonstration using Test Pattern 14 and Test Pattern 19
89	11.2	The Interrogator shall provide integrity protection for the command in the CryptoCommCmd payload (see Table 16).	M	Interrogator	By design
90	11.2	Integrity of the Interrogator command shall be performed with the addition of an 8-bit value of 00_h followed by a MAC.	M	Interrogator	By design
91	11.2	The **MA.2** state shall process crypto commands from the Tag's air interface protocol logic only when **ERROR**=FALSE.	M	Tag	By design
92	11.2	The Tag shall check the crypto command for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **MA.2** state.	M	Tag	By design
93	11.2	If secure authenticated communication is not enabled, an error condition shall occur for any CryptoAuthCmd, CryptoKeyUpdate or CryptoSecCommCmd in the **MA.2** state.	M	Tag	By design
94	11.2	If no error condition exists, the Tag shall generate a MAC for the Interrogator command within the CryptoCommCmd payload to authenticate the Interrogator command.	M	Tag	By design
95	11.2	The Tag accepts the Interrogator command as valid if the MAC from the Interrogator equals the MAC from the Tag.	M	Tag	By design
96	11.2	If the Interrogator command is invalid, the Tag crypto engine shall set **ERROR**=TRUE and the Tag shall remain in the **MA.2** state.	M	Tag	By design
97	11.2	The **MA.2** state shall also process crypto responses from the Tag's air interface protocol logic only when **ERROR**=FALSE.	M	Tag	By design
98	11.2	The Tag shall check the crypto command for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **MA.2** state.	M	Tag	By design
99	11.2	If the Tag responds to CryptoCommCmd, an error condition shall occur for any CryptoSecCommResp or CryptoAuthResp in the **MA.2** state.	M	Tag	By design
100	11.2	If the no error condition exists, the Tag shall provide integrity protection for the Tag response in the CryptoCommResp payload (see Table 17).	M	Tag	By design
101	11.2	Integrity of the Tag response shall be performed with the addition of an 8-bit value of 00_h followed by a MAC.	M	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
102	11.2	The Tag shall remain in the **MA.2** state after the response is sent to the Interrogator.	O	Tag	By design
103	11.2	The Interrogator shall generate a MAC for the Tag response within the CryptoCommResp payload to authenticate the Tag response.	O	Interrogator	By design
104	11.2	The Interrogator accepts the response as valid if the MAC from the Tag equals the MAC from the Interrogator.	O	Interrogator	By design
105	11.3	If a Tag and Interrogator are both authenticated as a result of mutual authentication, then the Interrogator has the option to enable the use of encrypted commands and responses when secure authenticated communication is supported by the Tag.	O	Interrogator, Tag	By demonstration using Test Pattern 20
106	11.3	Secure authenticated communication shall be enabled via the Options parameter during the mutual authentication process.	O	Interrogator	By design
107	11.3	The Interrogator shall encrypt the encapsulated Interrogator command in the CryptoSecCommCmd payload.	O	Interrogator	By design
108	11.3	The Interrogator shall provide integrity protection for the encrypted command in the CryptoSecCommCmd payload (see Table 18).	O	Interrogator	By design
109	11.3	Integrity of the Interrogator command shall be performed with the addition of an 8-bit value of 00_h followed by a MAC.	O	Interrogator	By design
110	11.3	The **MA.2** state shall process crypto commands and responses from the Tag's air interface protocol logic only when **ERROR**=FALSE.	O	Tag	By design
111	11.3	The Tag shall check the crypto responses for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **MA.2** state.	O	Tag	By design
112	11.3	An error condition shall occur for any CryptoAuthCmd or for any CrypteSecCommCmd when secure authenticated communication is not enabled.	O	Tag	By demonstration using Test Pattern 21
113	11.3	If no error condition, the Tag shall decrypt the Interrogator command within the CryptoSecCommCmd payload.	O	Tag	By design
114	11.3	The Tag shall generate a MAC for the Interrogator command within the CryptoSecCommCmd payload to authenticate the Interrogator command.	O	Tag	By design
115	11.3	The Tag accepts the Interrogator command as valid if the MAC from the Interrogator equals the MAC from the Tag.	O	Tag	By design
116	11.2	If the Interrogator command is invalid, the Tag crypto engine shall set **ERROR**=TRUE and the Tag shall remain in the **MA.2** state.	O	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
117	11.3	The Tag shall check the crypto responses for any error conditions. If an error condition exists then the Tag crypto engine shall set **ERROR**=True and remain in the **MA.2** state.	M	Tag	By design
118	11.3	An error condition shall occur for any CryptoAuthResp or for any CrypteSecCommResp when secure authenticated communication is not enabled.	M	Tag	By design
119	11.3	If no error condition exists, the Tag shall encrypt the response.	O	Tag	By design
120	11.3	The Tag shall provide integrity protection for the encrypted response in the CryptoSecCommResp payload (see Table 19).	O	Tag	By design
121	11.3	Integrity of the Tag response shall be performed with the addition of an 8-bit value of 00_h followed by a MAC.	O	Tag	By design
122	11.3	The Tag shall remain in the **MA.2** state after the response is sent to the Interrogator.	O	Tag	By design
123	11.3	The Interrogator shall decrypt the Tag response within the CryptoSecCommResp payload.	O	Interrogator	By design
124	11.3	The Interrogator shall generate a MAC for the Tag response within the CryptoSecCommResp payload to authenticate the Tag response.	O	Interrogator	By design
125	11.3	The Interrogator shall accept the response as valid if the MAC from the Tag equals the MAC from the Interrogator.	O	Interrogator	By design
126	12	Tags may implement an optional key table for storage of the crypto keys. If implemented Tags may permit an Interrogator to perform a key update in the key table using secure authenticated communication.	O	Interrogator, Tag	By demonstration using Test Pattern 25
127	12	The Interrogator shall provide the crypto key value as defined in Table 20.	O	Interrogator, Tag	By design
128	Annex A	Any combinations of Start States and Transitions not listed in Tables A.1 to A.7 shall result in an error and consequently cause the state machine to stay in its current state.	M	Tag	By design
129	E.1	The crypto suites that are defined by ISO/IEC 29167 can be defined by their Crypto Suite Identifier (CSI). According to ISO/IEC 29167-1 the CSS for this crypto suite shall be defined as the 6-bit value 000011_2. For use by the air interface protocols in this Annex the value is expanded to the 8-bit value 03_h.	M	Interrogator, Tag	By design
130	E.3.1	Interrogators and Tags implementing the Grain-128A CS shall provide the security services shown in Table E.1 using the protocol commands shown in Table E.3.	M	Interrogator, Tag	By design
131	E.3.1	During authentication, Tags shall report the features implemented in support of the Grain-128A CS as shown in Table E.2.	M	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
132	E.3.1	Tags shall implement both Tag authentication and mutual authentication and may implement Interrogator authentication.	M	Tag	By design
133	E.3.1	Authenticated communication shall be supported for all provided authentication methods.	M	Interrogator, Tag	By design
134	E.3.1	Secure authenticated communication may be supported for each provided authentication method.	O	Interrogator, Tag	By design
135	E.3.1	The Challenge command shall be implemented for all provided authentication methods.	M	Interrogator, Tag	By demonstration using Test Pattern 6 and Test Pattern 23
136	E.3.1	The Authenticate command shall be implemented for all provided authentication methods.	M	Interrogator, Tag	By demonstration using Test Pattern 1, Test Pattern 8 (only if IA is supported) and Test Pattern 14
137	E.3.1	The Tag execution time for authentication shall be less than 5ms.	M	Tag	By demonstration using Test Pattern 1, Test Pattern 8 (only if IA is supported) and Test Pattern 14
138	E.3.1	The AuthComm command shall be implemented for Authenticated communication.	M	Interrogator, Tag	By demonstration using Test Pattern 4, Test Pattern 12 (only if IA is supported) and Test Pattern 18
139	E.3.1	The SecureComm command shall be implemented if Secure authenticated communication is supported.	O	Interrogator, Tag	By demonstration using Test Pattern 20
140	E.3.1	The Tag shall support sending the contents of the response buffer in the reply to an ACK command.	M	Tag	By demonstration using Test Pattern 7 and Test Pattern 24
141	E.3.1	The Tag shall support sending the contents of the response buffer in the reply to a ReadBuffer command.	M	Tag	By demonstration using Test Pattern 5 and Test Pattern 22
142	E.3.1	The Tag may support a security timeout following a crypto error.	O	Tag	By design
143	E.3.1	A Tag shall not reply to a command having an invalid handle or invalid CRC, the crypto engine shall be reset and the next Tag state shall be **Open**.	M	Tag	By design
144	E.3.1	A crypto error shall reset the crypto engine, the Tag shall reply with the error code defined in section E.3.3 and the next Tag state shall be **Arbitrate**.	M	Tag	By design
145	E.3.1	For a successful Tag authentication the next Tag state is **Open**. For a successful Interrogator authentication or mutual authentication the next Tag state is **Secured**.	M	Tag	By design

Table 2 *(continued)*

Item	Protocol Sub clause	Requirement	MO	Applies to	How verified
146	E.3.1	The KeyUpdate command may be implemented.	O	Interrogator, Tag	By demonstration using Test Pattern 25
147	E.3.1	KeyUpate shall be supported only with encapsulation.	O	Interrogator, Tag	By design
148	E.3.3	Crypto Suite error conditions that may be reported to the Interrogator shall use the error codes specified in Table E.13.	M	Tag	By design

6.3 Test patterns

Test_Pattern 1

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b00, Options, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)

The selected value for Options shall be supported by the Tag. The crypto key used by the test system shall match with the selected crypto key of the Tag. The test pattern passed if the Tag provides a valid response to Authenticate and if TKeystream is equal to IKeystream. The authentication time shall be below 5 ms.

Test_Pattern 2

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b00, Options, KeyID, IRandomNumber=0xE79A6980CDB0)

The selected value for Options shall be supported, while the selected KeyID shall be not supported by the Tag. The test pattern passed if the Tag responds with an Error message.

Test_Pattern 3

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b01, Options, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)

The selected value for Options shall be supported by the Tag. The test pattern passed if the Tag responds with an Error message.

Test_Pattern 4

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b00, Options, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)
AuthComm (IncRepLen=0b0; Message=(Read(MemBank=0b01; WordPtr=0x01; WordCount=0x01))

The selected value for Options shall be supported by the Tag. The crypto key used by the test system shall match with the selected crypto key of the Tag. The test pattern passed if the Tag provides a valid response to AuthComm, encapsulating a valid response to the Read command. The provided MAC shall be valid.

Test_Pattern 5

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b0; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b00, Options, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)
ReadBuffer (RFU=0b00; WordPtr=0x000; BitCount=0x000)

The selected value for Options shall be supported by the Tag. The test pattern passed if no response field is included in the tag reply to Authenticate and if the response field of the Tag reply to ReadBuffer includes a valid TKeyStream.

Test_Pattern 6

Challenge (IncRepLen=0b0; Immed=0b0; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b00, Options, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)
Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
ReadBuffer (RFU=0b00; WordPtr=0x000; BitCount=0x000)

The selected value for Options shall be supported by the Tag. The test pattern passed if the Tag is not responding to Challenge and if the response field of the Tag reply to ReadBuffer includes a valid TKeyStream.

Test_Pattern 7

Challenge (IncRepLen=0b0; Immed=0b1; CSI=0x03; Length; Message=(AuthMeth=0b00, Step=0b00, Options, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)
Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK

The selected value for Options shall be supported by the Tag. The test pattern passed if the tag is not responding to Challenge and if the content of the response buffer is concatenated to the reply to ACK. The response field from the response buffer shall include a valid TKeyStream.

Test_Pattern 8

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b01, Step=0b00, Options=0b0000, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)
Authenticate (SenRep=1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b01, Step=0b01, Options, KeyID=0x00, IKeystream)

For the second Authenticate command, the selected value for Options shall be supported by the Tag. IKeystream shall be generated using a valid Key. The test pattern passed if the Tag responds with IA Status=0b0 to the second Authenticate. The authentication time shall be below 5 ms.

Test_Pattern 9

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b01, Step=0b00, Options=0b0001, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)

The test pattern passed if the Tag responds with an Error message.

Test_Pattern 10

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b01, Step=0b00, Options=0b0000, KeyID, IRandomNumber=0xE79A6980CDB0)

The selected value for KeyID shall be not supported by the Tag. The test pattern passed if the Tag responds with an Error message.

Test_Pattern 11

Query (Tari=12,5µs; BLF=320; Miller4, S0)
ACK
Req_RN
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b01, Step=0b00, Options=0b0000, KeyID=0x00, IRandomNumber=0xE79A6980CDB0)
Authenticate (SenRep=0b1; IncRepLen=0b0; CSI=0x03; Length; Message=(AuthMeth=0b01, Step=0b01, Options, KeyID=0x00, IKeystream)

For the second Authenticate command, the selected value for Options shall be supported by the Tag. IKeystream shall be generated using an invalid Key. The test pattern passed if the Tag responds with IA Status=0b1 to the second Authenticate.