EPC Compliant Class-1 Generation-2 UHF RFID Devices Conformance Requirements

specifies the conformance requirements for a passive-backscatter, Interrogator-talks-first, RFID system operating in the 860 – 960 MHz frequency range.

Release 2.0.1, Ratified, Oct 2015

Document Summary

Document Item	Current Value
Document Name	EPC Compliant Class-1 Generation-2 UHF RFID Devices Conformance Requirements
Document Date	Oct 2015
Document Version	2.0
Document Issue	1
Document Status	Ratified
Document Description	specifies the conformance requirements for a passive-backscatter, Interrogator-talks-first, RFID system operating in the 860 – 960 MHz frequency range.

Revision History

Release	Date of Change	Changed By	Summary of Change
2.0	Jun-2015	Josef Preishuber-Pflügl	Final Document
2.0.1	Oct 2015	D.Buckley	Branding update for ratified version

Disclaimer

GS1®, under its IP Policy, seeks to avoid uncertainty regarding intellectual property claims by requiring the participants in the Work Group that developed this **EPC Compliant Class-1 Generation-2 UHF RFID Devices Conformance Requirements** to agree to grant to GS1 members a royalty-free licence or a RAND licence to Necessary Claims, as that term is defined in the GS1 IP Policy. Furthermore, attention is drawn to the possibility that an implementation of one or more features of this Specification may be the subject of a patent or other intellectual property right that does not involve a Necessary Claim. Any such patent or other intellectual property right is not subject to the licencing obligations of GS1. Moreover, the agreement to grant licences provided under the GS1 IP Policy does not include IP rights and any claims of third parties who were not participants in the Work Group.

Accordingly, GS1 recommends that any organisation developing an implementation designed to be in conformance with this Specification should determine whether there are any patents that may encompass a specific implementation that the organisation is developing in compliance with the Specification and whether a licence under a patent or other intellectual property right is needed. Such a determination of a need for licencing should be made in view of the details of the specific system designed by the organisation in consultation with their own patent counsel.

THIS DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGMENT, FITNESS FOR PARTICULAR PURPOSE, OR ANY WARRANTY OTHER WISE ARISING OUT OF THIS SPECIFICATION. GS1 disclaims all liability for any damages arising from use or misuse of this Standard, whether special, indirect, consequential, or compensatory damages, and including liability for infringement of any intellectual property rights, relating to use of information in or reliance upon this document.

GS1 retains the right to make changes to this document at any time, without notice. GS1 makes no warranty for the use of this document and assumes no responsibility for any errors which may appear in the document, nor does it make a commitment to update the information contained herein.

GS1 and the GS1 logo are registered trademarks of GS1 AISBL.

Table of Contents

Introduction

This document specifies the conformance requirements for a passive-backscatter, Interrogator-talks-first (ITF), radio-frequency identification (RFID) system operating in the 860 MHz – 960 MHz frequency range. The system comprises Interrogators, also known as Readers, and Tags, also known as Labels.

An Interrogator transmits information to a Tag by modulating an RF signal in the 860 MHz – 960 MHz frequency range. The Tag receives both information and operating energy from this RF signal. Tags are passive, meaning that they receive all of their operating energy from the Interrogator's RF waveform.

An Interrogator receives information from a Tag by transmitting a continuous-wave (CW) RF signal to the Tag; the Tag responds by modulating the reflection coefficient of its antenna, thereby backscattering an information signal to the Interrogator. The system is ITF, meaning that a Tag modulates its antenna reflection coefficient with an information signal only after being directed to do so by an Interrogator.

Interrogators and Tags are not required to talk simultaneously; rather, communications are half-duplex, meaning that Interrogators talk and Tags listen, or vice versa.

1 Scope

This document specifies:

- Compliance requirements for physical interactions (the signalling layer of the communications) between Interrogators and Tags, and
- Compliance requirements for Interrogator and Tag operating procedures and commands.

2 Conformance

2.1 Claiming Conformance

A device shall not claim conformance with the Protocol unless certified, in writing, by EPCglobal, Inc., or one of its designated representatives. To conform, a device shall comply with all clauses in this document (except those marked as optional) and all local radio regulations. Conformance may also require a license from the owner of any intellectual property utilised by said device.

To be certified as alteration-EAS conformant Tags and Interrogators shall additionally support the optional clauses or portions of optional clauses specified in Annex N under the item Alteration EAS

To be certified consumer-electronics conformant Tags and Interrogators shall additionally support the optional clauses or portions of optional clauses specified in Annex N under the item Consumer Electronics

To be certified as Tag-alteration conformant and Interrogators shall additionally support the optional clauses or portions of optional clauses specified in Annex N under the item Tag Alteration (Core). Additionally tags may be conformant to Tag Alteration (Challenge), Tag Alteration (Authenticate) and or Tag Alteration (Full) by fulfilling the respective requirements of Annex N

2.2 General Conformance Requirements

2.2.1 Interrogators

To conform to the Protocol, an Interrogator shall:

- Meet the requirements of the Protocol,
- Implement the mandatory commands defined in the Protocol,

- Modulate/transmit and receive/demodulate a sufficient set of the electrical signals defined in the signaling layer of the Protocol to communicate with conformant Tags, and
- Conform to all local radio regulations.

To conform to the Protocol, an Interrogator may:

- Implement any subset of the optional commands defined in the Protocol, and
- Implement any proprietary and/or custom commands in conformance with the Protocol.

To conform to the Protocol, an Interrogator shall not:

- Implement any command that conflicts with the Protocol, or
- Require using an optional, proprietary, or custom command to meet the requirements of the Protocol.

2.2.2 Tags

To conform to the Protocol, a Tag shall:

- Meet the requirements of the Protocol,
- Implement the mandatory commands defined in the Protocol,
- Modulate a backscatter signal only after receiving the requisite command from an Interrogator, and
- Conform to all local radio regulations when appropriately commanded by an Interrogator.

To conform to the Protocol, a Tag may:

- Implement any subset of the optional commands defined in the Protocol, and
- Implement any proprietary and/or custom commands as defined in 2.3.3 and 2.3.4, respectively.

To conform to the Protocol, a Tag shall not:

- Implement any command that conflicts with the Protocol,
- Require using an optional, proprietary, or custom command to meet the requirements of the Protocol, or
- Modulate a backscatter signal unless commanded to do so by an Interrogator using the signaling layer defined in the Protocol.

2.3 Command Structure and Extensibility

Subclause 6.3.2.10 of the Protocol defines the structure of the command codes used by Interrogators and Tags, as well as the availability of future extensions. Each command is defined and labeled as mandatory or optional.

2.3.1 Mandatory Commands

Conforming Tags and Interrogators shall support all mandatory commands.

2.3.2 Optional Commands

Conforming Interrogators may or may not support optional commands. Conforming Tags may or may not support optional commands. If an Interrogator or a Tag implements an optional command, it shall implement it in the manner specified.

2.3.3 Proprietary Commands

Proprietary commands may be enabled in conformance with the Protocol, but are not specified in the Protocol. All proprietary commands shall be capable of being permanently disabled. Proprietary commands are intended for manufacturing purposes and shall not be used in field-deployed RFID systems.

2.3.4 Custom Commands

Custom commands may be enabled in conformance with the Protocol, but are not specified in the Protocol. An Interrogator shall issue a custom command only after singulating a Tag and reading (or having prior knowledge of) the Tag manufacturer's identification in the Tag's TID memory. An Interrogator shall use a custom command only in accordance with the specifications of the Tag manufacturer identified in the TID.

3 Normative References

The following referenced documents are indispensable to the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition (including any amendments) applies.

- **EPCglobal™:** *EPC™ Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID, Protocol for Communications at 860 MHz – 960 MHz, Version 2.0.0*
- **EPCglobal™:** *EPC™ Tag Data Standards*
- **European Telecommunications Standards Institute (ETSI), EN 302 208:** *Electromagnetic compatibility and radio spectrum matters (ERM) – Radio-frequency identification equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W, Part 1 – Technical characteristics and test methods*
- **European Telecommunications Standards Institute (ETSI), EN 302 208:** *Electromagnetic compatibility and radio spectrum matters (ERM) – Radio-frequency identification equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W, Part 2 – Harmonized EN under article 3.2 of the R&TTE directive*
- **ISO / IEC Directives, Part 2:** *Rules for the structure and drafting of International Standards*
- **ISO / IEC 3309:** *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame structure*
- **ISO / IEC 18000-63:** Information technology automatic identification and data capture techniques — Radio frequency identification for item management air interface — Part 63: Type C: Parameters for air interface communications at 860–960 MHz
- **ISO / IEC 19762:** *Information technology AIDC techniques – Harmonized vocabulary – Part 3: radio-frequency identification (RFID)*
- ***U.S. Code of Federal Regulations (CF*R), Title 47, Chapter I, Part 15:** *Radio-frequency devices, U.S. Federal Communications Commission*

4 Terms and Definitions

The principal terms and definitions used in this document are described in the Protocol and in ISO/IEC 19762.

4.1 Additional Terms and Definitions

Terms and definitions specific to this document that supersede any normative references are as follows:

- **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 approved by EPCglobal™ shall certify the technical analysis as being 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, except in special cases, 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.

- **By demonstration**

  Laboratory testing of one, or if required for statistical reasons multiple, products, processes, or services to ensure compliance. A test laboratory certified by EPCglobal™ 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 certifying test laboratory.

  Interrogators submitted for testing purposes shall include physical connections and test modes suitable for the certifying laboratory to evaluate Interrogator performance under the test conditions specified in this document.

  Tags submitted for testing purposes shall include all documentation required by 6.3.1.3.5 of the Protocol. The certifying laboratory's test plan will specify the submitted Tag's memory contents (i.e. the contents of Reserved, EPC, TID, and User memory as well as the lock status of these memory banks).

- **As implemented**

  If a Tag or Interrogator implements a subset of the Protocol, compliance shall be verified over the subset actually implemented. For example, although Interrogators may implement DSB-ASK, SSB-ASK, or PR-ASK modulation, a manufacturer may choose to only implement DSB-ASK modulation, in which case compliance testing shall only use DSB-ASK modulation. For parameters that are continuously variable, compliance shall be verified at the minimum and maximum values of the implemented range, unless the test conditions specifically state otherwise.

5 Symbols, Abbreviated Terms, and Notation

The principal symbols and abbreviated terms used in this document are detailed in

- **ISO / IEC 19762:** *Information technology AIDC techniques – vocabulary*
- **EPCglobal™:** *EPC™ Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID, Protocol for Communications at 860 MHz – 960 MHz, Version 2.0.0*

Symbols, abbreviated terms, and notation specific to this document are as follows:

5.1 Symbols

None

5.2 Abbreviated Terms

None

5.3 Notation

This document uses the following notational conventions:

- States and flags are denoted in bold. Example: **ready**.
- Commands are denoted in italics. Variables are also denoted in italics. Where there might be confusion between commands and variables, this specification will make an explicit statement. Example: *Query*.
- Command parameters are underlined. Example: Pointer.
- For logical negation, labels are preceded by '~'. Example: If **flag** is true, then **~flag** is false.
- The symbol, R=>T, refers to commands or signalling from an Interrogator to a Tag (Reader-to-Tag).
- The symbol, T=>R, refers to commands or signalling from a Tag to an Interrogator (Tag-to-Reader).

6 Protocol Requirements

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

- M mandatory
- EAS mandatory for EAS only, otherwise optional
- TAC mandatory for Tag Alteration (Core), otherwise optional
- TACC mandatory for Tag Alteration (Core + Challenge), otherwise optional
- TACA mandatory for Tag Alteration (Core + Authenticate), otherwise optional
- TACF mandatory for Tag Alteration (Core + Full), otherwise optional
- CE mandatory for Consumer Electronics, otherwise optional
- *O* *optional*
- *C* *required in respect to crypto suite*

To be certified as alteration-EAS, Tag-alteration, and/or consumer-electronics conformant, Tags and Interrogators shall support the optional clauses or portions of optional clauses cited in the corresponding Table 6-1, respectively, as mandatory. To be clear, those features in the cited optional clause or portion of optional clause specified with a "may" shall become a "shall"; those specified with a "shall" shall remain a "shall".

Unless otherwise specified, testing shall take place in an environment of temperature 23° C +/- 3° C and of relative humidity 40 % to 60 %.

Table 6-1 Protocol Requirement

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
1	6.1.1	Tags shall not be required to demodulate Interrogator commands while backscattering.	M	Tag	**By design**
2	6.1.1	A Tag shall not respond to a mandatory or optional command using full-duplex communications.	M	Tag	**By design**
3	6.3.1.1	Tags shall receive power from and communicate with Interrogators within the frequency range from 860 MHz to 960 MHz, inclusive.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 25 µs RTcal: 62.5 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRcal: 100 µs DR: 8 M: 1 TRext: 0
4	6.3.1.1	Interrogators certified for operation in dense-Interrogator environments shall support, but are not required to always use, the dense-Interrogator mode described in *Annex G*.	M	Interro-gator	**By design**
5	6.3.1.2	Interrogators shall use a fixed modulation format and data rate for the duration of an inventory round, where “inventory round” is defined in 4.1	M	Interro-gator	**By design**
6	6.3.1.2.1	Interrogators certified for operation in single- or multiple-Interrogator environments shall have a frequency accuracy that meets local regulations.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
7	6.3.1.2.1	Interrogators certified for operation in dense-Interrogator environments shall have a frequency accuracy of +/– 10 ppm over the nominal temperature range (–25 °C to +40 °C) and +/– 20 ppm over the extended temperature range (–40°C to +65°C) while transmitting, unless local regulations specify tighter accuracy, in which case the Interrogator frequency accuracy shall meet the local regulations.	M	Interro-gator	**By demonstration**, for dense-Interrogator certification, unless local regulations specify tighter frequency accuracy than the Protocol, in which case the Interrogator manufacturer shall provide evidence of certification by the local regulatory body in lieu of laboratory demonstration. Test conditions: Temp: max(–40, minimum supported temperature) and min(65, maximum supported temperature). If supported temperature range exceeds –25 or 40 then testing shall also be performed at –25 or 40 respectively. All temperatures are in ºC (all +/– 3 ºC). See Annex A, Q7. Freq: 5 test points situated at the band edges and linearly spanning the supported band at valid channel frequencies. Test can be skipped if regulatory approval shows that these requirements have been fulfilled already. Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: 1 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation method: - continuous wave to allow exact measurement based on the clear peak
8	6.3.1.2.2	Interrogators shall communicate using DSB-ASK, SSB-ASK, or PR-ASK modulation, detailed in *Annex H*.	M	Interro-gator	**By design**
9	6.3.1.2.2	Tags shall demodulate all three modulation types.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK, SSB-ASK, & PR-ASK Tari: 6.25, 12.5, & 25 µs RTcal: 2.5×Tari PW: min and max Modulation depth: 90% ASK, 200% PR-ASK DSB-ASK rise/fall time: ≤ 0.33 Tari SSB-ASK rise/fall time: ≤ 0.33 Tari PR-ASK rise/fall time: ≤ 0.62×PW TRcal: 2×RTcal DR: 8 M: 1 TRext: 0
10	6.3.1.2.3	The R=>T link shall use PIE, shown in Figure 6.1.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
11	6.3.1.2.3	Pulse modulation depth, rise time, fall time, and PW shall be as specified in Table 6.5, and shall be the same for a data-0 and a data-1.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented See Annex A, Q10.
12	6.3.1.2.3	Interrogators shall use a fixed modulation depth, rise time, fall time, PW, Tari, data-0 length, and data-1 length for the duration of an inventory round.	M	Interro-gator	**By design**
13	6.3.1.2.3	The RF envelope shall be as specified in Figure 6.2.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented See Annex A, Q10.
14	6.3.1.2.4	Interrogators shall communicate using Tari values in the range of 6.25µs to 25µs.	M	Interro-gator	**By design**
15	6.3.1.2.4	Interrogator compliance shall be evaluated using at least one Tari value between 6.25µs and 25µs with at least one value of the parameter *x*.	M	Interro-gator	This document uses vendor preferred Tari and x valvues as consistent with the Protocol.
16	6.3.1.2.4	The tolerance on all parameters specified in units of Tari shall be +/–1%.	M	Interro-gator	**By demonstration** Test conditions: Temp: Either (a) or (b) shown below a) Single and Multi-Interrogators: 23 ºC +/– 3 ºC b) Dense-Interrogators tested at modulation, data rate, and encoding parameters specified in Annex G of the Protocol specification: max(–40, minimum supported temperature) and min(65, maximum supported temperature). If supported temperature range exceeds –25 or 40 then testing shall also be performed at –25 or 40 respectively. All temperatures are in ºC (all +/– 3 ºC). See Annex A, Q7. Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented
17	6.3.1.2.4	The choice of Tari value and *x* shall be in accordance with local radio regulations.	M	Interro-gator	**By design**
18	6.3.1.2.5	The R=>T RF envelope shall comply with Figure 6.2 and Table 6.1.	M	Interro-gator	Tested in compliance with 6.3.1.2.3
19	6.3.1.2.5	An Interrogator shall not change the R=>T modulation type (i.e. shall not switch between DSB-ASK, SSB-ASK, or PR-ASK) without first powering down its RF waveform (see 6.3.1.2.7).	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
20	6.3.1.2.6	The Interrogator power-up RF envelope shall comply with Figure 6.3 and Table 6.6.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. See Annex A, Q8.
21	6.3.1.2.6	Once the carrier level has risen above the 10% level, the power-up envelope shall rise monotonically until at least the ripple limit M_I. The RF envelope shall not fall below the 90% point in Figure 6.3 during interval T_s.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. See Annex A, Q9.
22	6.3.1.2.6	Interrogators shall not issue commands before the end of the maximum settling-time interval in Table 6.6 (i.e. before T_s).	M	Interro-gator	**By design**
23	6.3.1.2.7	The Interrogator power-down RF envelope shall comply with Figure 6.3 and Table 6.7.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented.
24	6.3.1.2.7	Once the carrier level has fallen below the 90% level, the power-down envelope shall fall monotonically until the power-off limit M_s.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. See Annex A, Q9.
25	6.3.1.2.7	Once powered off, an Interrogator shall remain powered off for a least 1ms before powering up again.	M	Interro-gator	**By design**
26	6.3.1.2.8	An Interrogator shall begin all R=>T signaling with either a preamble or a frame-sync, both of which are shown in Figure 6.4.	M	Interro-gator	**By design**
27	6.3.1.2.8	A preamble shall precede a *Query* command (see 6.3.2.12.2.1) and denotes the start of an inventory round.	M	Interro-gator	**By design** – inherently tested with other tests
28	6.3.1.2.8	All other signaling shall begin with a frame-sync.	M	Interro-gator	**By design** – inherently tested with other tests
29	6.3.1.2.8	The tolerance on all parameters specified in units of Tari shall be +/–1%.	M	Interro-gator	Tested in compliance with 6.3.1.2.3
30	6.3.1.2.8	PW shall be as specified in Table 6.5.	M	Interro-gator	Tested in compliance with 6.3.1.2.3
31	6.3.1.2.8	The RF envelope shall be as specified in Figure 6.2.	M	Interro-gator	**By design**
32	6.3.1.2.8	A preamble shall comprise a fixed-length start delimiter, a data-0 symbol, an R=>T calibration (RTcal) symbol, and a T=>R calibration (TRcal) symbol.	M	Interro-gator	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
33	6.3.1.2.8	An Interrogator shall set RTcal equal to the length of a data-0 symbol plus the length of a data-1 symbol (RTcal = 0_{length} + 1_{length}).	M	Interro-gator	**By design**
34	6.3.1.2.8	A Tag shall measure the length of RTcal and compute *pivot* = RTcal / 2.	M	Tag	**By design**
35	6.3.1.2.8	A Tag shall interpret subsequent Interrogator symbols shorter than *pivot* to be data-0s, and subsequent Interrogator symbols longer than *pivot* to be data-1s.	M	Tag	**By design**
36	6.3.1.2.8	A Tag shall interpret symbols longer than 4 RTcal to be invalid.	M	Tag	**By design**
37	6.3.1.2.8	Prior to changing RTcal, an Interrogator shall transmit CW for a minimum of 8 RTcal.	M	Interro-gator	**By design**
38	6.3.1.2.8	An Interrogator shall specify a Tag's backscatter link frequency (its FM0 datarate or the frequency of its Miller subcarrier) using the TRcal and divide ratio (DR) in the preamble and payload, respectively, of a *Query* command that initiates an inventory round.	M	Interro-gator	**By design**
39	6.3.1.2.8	A Tag shall measure the length of TRcal, compute BLF, and adjust its T=>R link rate to be equal to BLF (Table 6.9 shows BLF values and tolerances).	M	Tag	Tested in compliance with 6.3.1.3.3
40	6.3.1.2.8	The TRcal and RTcal that an Interrogator uses in any inventory round shall meet the constraints in Equation (2) : 1.1 x RTcal ≤ TRcal ≤ 3 x RTcal	M	Interro-gator	**By design** – inherently tested with other tests
41	6.3.1.2.8	An Interrogator, for the duration of an inventory round, shall use the same length RTcal in a frame-sync as it used in the preamble that initiated the round.	M	Interro-gator	**By design**
42	6.3.1.2.9	When an Interrogator uses frequency-hopping spread spectrum (FHSS) signaling, the Interrogator's RF envelope shall comply with Figure 6.5 and Table 6.8. The RF envelope shall not fall below the 90% point in Figure 6.5 during interval T_{hs}.	M	Interro-gator	**By demonstration**, for Interrogators that use FHSS: Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented.
43	6.3.1.2.9	Interrogators shall not issue commands before the end of the maximum settling-time interval in Table 6.8 (i.e. before T_{hs}).	M	Interro-gator	**By design**
44	6.3.1.2.9	The maximum time between frequency hops and the minimum RF-off time during a hop shall meet local regulatory requirements.	M	Interro-gator	**By design**
45	6.3.1.2.10	Interrogators certified for operation in single-Interrogator environments shall meet local regulations for spread-spectrum channelization.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
46	6.3.1.2.10	Interrogators certified for operation in multiple- or dense-Interrogator environments shall meet local regulations for spread-spectrum channelization, unless the channelization is unregulated, in which case Interrogators shall adopt the channel plan at *http://www.gs1.org/epcglobal/implementation* for the chose regulatory region (see also *Annex G*, which describes multiple- and dense-Interrogator channelized signaling).	M	Interro-gator	**By demonstration**, for multiple- or dense-Interrogator certification. Test conditions: Temp: 23 +/– 3 ºC Freq: Either (a) or (b) shown below a) Interrogators that are capable of commanding Tags to backscatter using subcarrier signaling: 50 discrete center frequencies as specified in Table G.1 of the Protocol. b) Interrogators that are not capable of commanding Tags to backscatter using subcarrier signaling: All center frequencies supported by the Interrogator (note: the certification laboratory reserves the right to test a random subset of the Interrogator's supported center frequencies). Power: Maximum Interrogator transmit power, as implemented Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: 1 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation method: - continuous modulation Test can be skipped if regulatory approval shows that these requirements have been fulfilled already.
47	6.3.1.2.11	Interrogators certified for operation according to this protocol shall meet local regulations for out-of-channel and out-of-band spurious radio-frequency emissions.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
48	6.3.1.2.11	Interrogators certified for operation in multiple-Interrogator environments, shall meet both local regulations,and the Multiple-Interrogator Transmit Mask described below and shown in Figure 6.6..	M	Interro-gator	**By demonstration**, for multiple-Interrogator certification. Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Channel width: 200 kHz for Interrogators certified for operation in Europe; A maximum of 500 kHz for Interrogators certified for operation in North America. Modulation: As implemented Transmit data: Either (a) or (b), below a) a continuous repeating 9-bit maximum length sequence with polynomial $x^9 + x^4 + 1$, initially seeded with all ones, resulting in a repeating 511-bit sequence of FF83DF1732094ED1E7CD8A91C6D5C4C44021184E5586F4DC8A15A7EC92DF93533018CA34BFA2C759678FBA0D6DD82D7D540A57977039D27AEA243385ED9A1DE0$_h$, or b) a repeating *Select* command with a 252 bit Mask value set to ACBCD2114DAE1577C6DBF4C91A3CDA2F169B340989C1D32C290465E5C1423CC$_h$ Bit sequences are listed MSB first. Other transmit parameters: As implemented Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: 1 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation method: - continuous modulation
49	6.3.1.2.11	Multiple-Interrogator Transmit Mask: For an Interrogator transmitting random data in channel *R*, and any other channel *S*≠*R*, the ratio of the integrated power *P*() in channel *S* to that in channel *R* shall not exceed the specified values:	M	Interro-gator	Tested in compliance with 6.3.1.2.11, Figure 6.6
50	6.3.1.2.11	Each channel that exceeds the mask shall be counted as an exception.	M		Tested in compliance with 6.3.1.2.11, Figure 6.6

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
51	6.3.1.2.11	Interrogators certified for operation in dense-Interrogator environments shall meet both local regulations and the Dense-Interrogator Transmit Mask described below and shown in Figure 6.7.	M	Interro-gator	**By demonstration**, for dense-Interrogator certification. Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Reference bandwidth: 2.5/Tari Modulation: As implemented Transmit data: continuous modulation Tari: according to vendor-selected value for normal operation Backscatter data rate: One or more of the dense-interrogator data rates specified in Annex G of the Protocol specification, as implemented. Other transmit parameters: As implemented Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: according to Tari ranges below Tari >=20 µs: 1 MHz 20 µs > Tari >= 10 µs: 2 MHz Tari < 10 µs: 3 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation method: - continuous modulation
52	6.3.1.2.11	Regardless of the mask used, Interrogators certified for operation in dense-Interrogator environments shall not be permitted the two exceptions to the transmit mask that are allowed for Interrogators certified for operation in multiple-Interrogator environments.	M	Interro-gator	Tested in compliance with 6.3.1.2.11, Figure 6.7 Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: 1 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation method: - continuous repeated inventory sequence (no tags present)
53	6.3.1.2.11	For Interrogator transmissions centered at a frequency f_c, a 2.5/Tari bandwidth R_{BW} also centered at f_c, an offset frequency f_o = 2.5/Tari, and a 2.5/Tari bandwidth S_{BW} centered at $(n \times f_o) + f_c$ (integer n), the ratio of the integrated power $P()$ in S_{BW} to that in R_{BW} with the Interrogator transmitting random data shall not exceed the specified values:	M	Interro-gator	Tested in compliance with 6.3.1.2.11, Figure 6.7 Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: 1 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation method: - continuous repeated inventory sequence (no tags present)
54	6.3.1.3	A Tag shall backscatter using a fixed modulation format, data encoding, and data rate for the duration of an inventory round, where “inventory round” is defined in 4.1.	M	Tag	**By design**
55	6.3.1.3.1	Tag backscatter shall use ASK and/or PSK modulation.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
56	6.3.1.3.1	Interrogators shall demodulate both modulation types.	M	Interro-gator	**By design**
57	6.3.1.3.2	Tags shall encode the backscattered data as either FM0 baseband or Miller modulation of a subcarrier at the data rate.	M	Tag	Tested in compliance with 6.3.1.3.2.1 and 6.3.1.3.2.3
58	6.3.1.3.2.1	The duty cycle of a 00 or 11 sequence, measured at the modulator output, shall be a minimum of 45% and a maximum of 55%, with a nominal value of 50%.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRext: 0 Test # 1 Tari: 6.25 µs RTcal: 18.75 µs TRcal: 33.3 & 50 µs DR: 64/3 M: 1 Test # 2 Tari: 12.5 µs RTcal: 31.25 µs TRcal: 66.7, 83.3 µs DR: 64/3 M: 1
59	6.3.1.3.2.1	FM0 signaling shall always end with a “dummy” data-1 bit at the end of a transmission, as shown in Figure 6.10.	M	Tag	**By design**
60	6.3.1.3.2.2	T=>R FM0 signaling shall begin with one of the two preambles shown in Figure 6.11.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0 & 1
61	6.3.1.3.2.2	The choice depends on the TRext value specified in the *Query* that initiated the inventory round, unless a Tag is replying to a command that uses a delayed or in-process reply (see 6.3.1.6), in which case a Tag shall use the extended preamble regardless of TRext (i.e. the Tag replies as if TRext=1 regardless of the TRext value specified in the *Query*—see 6.3.2.12.3).	M	Tag	**By demonstration** Tested in compliance with 6.3.2.6, Figure 6.21
62	6.3.1.3.2.3	Figure 6.13 shows Miller-modulated subcarrier sequences; the Miller sequence shall contain exactly two, four, or eight subcarrier cycles per bit, depending on the M value specified in the *Query* command that initiated the inventory round (see Table 6.10).	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
63	6.3.1.3.2.3	The duty cycle of a 0 or 1 symbol, measured at the modulator output, shall be a minimum of 45% and a maximum of 55%, with a nominal value of 50%.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRext: 0 Test # 1 Tari: 6.25 µs RTcal: 18.75 µs TRcal: 33.3 & 50 µs DR: 64/3 M: 2, 4, 8 Test # 2 Tari: 12.5 µs RTcal: 31.25 µs TRcal: 66.7, 83.3 µs DR: 64/3 M: 2, 4, 8
64	6.3.1.3.2.3	Miller signaling shall always end with a “dummy” data-1 bit at the end of a transmission, as shown in Figure 6.14.	M	Tag	**By design**
65	6.3.1.3.2.4	T=>R subcarrier signaling shall begin with one of the two preambles shown in Figure 6.15.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 2, 4, 8 TRext: 0 & 1
66	6.3.1.3.2.4	The choice depends on the TRext value specified in the *Query* that initiated the inventory round, unless a Tag is replying to a command that uses a delayed or in-process reply (see 6.3.1.6), in which case a Tag shall use the extended preamble regardless of TRext (i.e. the Tag replies as if TRext=1 regardless of the TRext value specified in the *Query*—see 6.3.2.10.3).	M	Tag	**By demonstration** Tested in compliance with 6.3.2.6, Figure 6.21

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
67	6.3.1.3.3	Tags shall support all R=>T Tari values in the range of 6.25µs to 25µs, over all parameters allowed by 6.3.1.2.3. Tags shall support the T=>R link frequencies and tolerances specified in Table 6.9 and the T=>R data rates specified in Table 6.10.	M	Tag	The FT requirements in Table 6.9 of the Protocol shall be verified **by design**. Tag manufacturers shall provide plots of worst-case FT error versus TRcal. Tag manufacturers shall also provide measured data used to generate the FT plots, including: 1. Tag oscillator frequency tolerance 2. Tag oscillator frequency drift 3. TRcal measurement error budget 4. Other contributors to FT error The frequency-variation during backscatter requirements in Table 6.9 of the Protocol shall be verified **by demonstration**. The testing laboratory shall measure the minimum, median, and maximum symbol length (M=1) or subcarrier period (M=2, 4, 8) during backscatter of a 128-bit sequence (16-bit PC, 96-bit EPC, and a CRC-16). The minimum and maximum values shall not deviate by more than 2.5% from the median. The test conditions are: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRext: 0 Test # 1 Tari: 6.25 µs; RTcal: 18.75 µs TRcal: 33.3*0.99, 33.3 & 33.3*1.01 µs DR: 64/3; M: 1, 2, 4, 8 For all 3 Trcal values above FT shall be verified with the value for TRcal = 33.3 µs. Test # 2 Tari: 6.25 µs; RTcal: 18.75 µs TRcal: 50*0.99, 50 & 50 * 1.01 µs DR: 64/3; M: 1, 2, 4, 8 For all 3 Trcal values above FT shall be verified with the value for TRcal = 50 µs. Test # 3 Tari: 25 µs; RTcal: 75 µs TRcal: 200 * 0,99, 200 & 200 * 1.01 µs DR: 8; M: 1, 2, 4, 8 For all 3 Trcal values above FT shall be verified with the value for TRcal = 200 µs.
68	6.3.1.3.4	Tags energized by an Interrogator shall be capable of receiving and acting on Interrogator commands within a period not exceeding the maximum settling-time interval specified in Table 6.6 or Table 6.8, as appropriate (i.e. within T_s or T_{hs}, respectively).	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
69	6.3.1.3.5	For a Tag certified to this protocol, the Tag manufacturer shall specify: 1. free-space, sensitivity, 2: minimum backscattered modulated power (ASK modulation) or change in radar cross-section or equivalent (phase modulation), and 3: the manufacturer's normal operating conditions for the Tag mounted on one or more manufacturer-selected materials.	M	Tag	**By design** The Tag manufacturer shall provide this specification in the supplied documents.
70	6.3.1.4	The transmission order for all R=>T and T=>R communications shall be most-significant bit (MSB) first.	M	Tag and Interro-gator	**By design**
71	6.3.1.4	Within each message, the most-significant word shall be transmitted first.	M	Tag and Interro-gator	**By design**
72	6.3.1.4	Within each word, the MSB shall be transmitted first.	M	Tag and Interro-gator	**By design**
73	6.3.1.5	To generate a CRC-16 a Tag or Interrogator shall first generate the CRC-16 precursor shown in Table 6.11, and then take the ones-complement of the generated precursor to form the CRC-16.	M	Tag and Interro-gator	**By design**
74	6.3.1.5	A Tag or Interrogator shall verify the integrity of a received message that uses a CRC-16.	M	Tag and Interro-gator	**By design**
75	6.3.1.5	Tags shall append a CRC-16 to those replies that use a CRC-16 — see 6.3.2.12 for command-specific replies.	M	Tag	**By design**
76	6.3.1.5	To generate a CRC-5 an Interrogator shall use the definition in Table 6.12.	M	Interro-gator	**By design**
77	6.3.1.5	A Tag shall verify the integrity of a received message that uses a CRC-5.	M	Tag	**By design**
78	6.3.1.5	Interrogators shall append the appropriate CRC to R=>T transmissions as specified in Table 6.28.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
79	6.3.1.6	Tags and Interrogators that support Immediate Reply Type (see Table 6.28) shall meet all timing requirements shown in Table 6.16.	M	Tag and Interro-gator	**By demonstration** Interrogator test conditions: Verify Interrogator meets T_2, T_3, & T_4 Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented Tag test conditions: Verify Tag meets T_1 over T_2 extremes Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRext: 0 Minimum T_2 condition: Tari: 6.25 µs RTcal: 18.75 µs TRcal: 33.3 & 50 µs DR: 64/3 M: 1 Maximum T_2 condition: Tari: 25 µs RTcal: 75 µs TRcal: 200 µs DR: 8 M: 2, 4, 8
80	6.3.1.6	Tags and Interrogators that support Delayed Reply Type (see Table 6.28) shall meet all timing requirements shown in Table 6.16.		Tag and Interro-gator	**By demonstration** Interrogator test conditions: Verify Interrogator meets T_2, & T_4 Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented Tag test conditions: Verify Tag meets T_5 over T_2 extremes Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRext: 1 Minimum T_2 condition: Tari: 6.25 µs RTcal: 18.75 µs TRcal: 33.3 & 50 µs DR: 64/3 M: 1 Maximum T_2 condition: Tari: 25 µs RTcal: 75 µs TRcal: 200 µs DR: 8 M: 2, 4, 8

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
81	6.3.1.6	Tags and Interrogators that support In-process Reply Type (see Table 6.28) shall meet all timing requirements shown in Table 6.16.	M	Tag and Interro-gator	**By demonstration** Interrogator test conditions: Verify Interrogator meets T_2 & T_4 Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Other transmit parameters: As implemented Tag test conditions: Verify Tag meets T_6 & T_7 over T_2 extremes Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari TRext: 1 Minimum T_2 condition: Tari: 6.25 µs RTcal: 18.75 µs TRcal: 33.3 & 50 µs DR: 64/3 M: 1 Maximum T_2 condition: Tari: 25 µs RTcal: 75 µs TRcal: 200 µs DR: 8 M: 2, 4, 8
82	6.3.1.6	As described in 6.3.1.2.8, an Interrogator shall use a fixed R=>T link rate for the duration of an inventory round.	M	Interro-gator	**By design**
83	6.3.1.6	Prior to changing the R=>T link rate, an Interrogator shall transmit CW for a minimum of 8 RTcal.	M	Interro-gator	**By design**
84	6.3.1.6 Table 6.16	The maximum value for T_2 shall apply only to Tags in the **reply** or **acknowledged** states (see 6.3.2.6.3 and 6.3.2.6.4).	M	Tag	**By demonstration** Issue a Read command in Secured State. T_2 before the Read command shall be at least 10 ms. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
85	6.3.1.6 Table 6.16	For a Tag in the **reply** or **acknowledged** states, if T_2 expires (i.e. reaches its maximum value) without the Tag receiving a valid command, the Tag shall transition to the **arbitrate** state (see 6.3.2.6.2).	M	Tag	**By Design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
86	6.3.1.6 Table 6.16	For a Tag in the **reply** or **acknowledged** states, if T_2 expires (i.e. reaches its maximum value) during the reception of a valid command, the Tag shall execute the command.	M	Tag	**By Design**
87	6.3.1.6 Table 6.16	For a Tag in the **reply** or **acknowledged** states, if T_2 expires (i.e. reaches its maximum value) during the reception of an invalid command, the Tag shall transition to **arbitrate** upon determining that the command is invalid.	M	Tag	**By Design**
88	6.3.1.6 Table 6.16	In all other states the maximum value for T_2 shall be unrestricted.	M	Tag	**By design**
89	6.3.1.6	T_1+T_3 shall not be less than T_4	M	Tag	**By design**
90	6.3.1.6.2	After issuing a command that uses delayed reply timing an Interrogator shall transmit CW for at least the lesser of T_{REPLY} or $T_{5(max)}$, where T_{REPLY} is the time between the Interrogator's command and the Tag's backscattered reply.	M	Interro-gator	**By demonstration**
91	6.3.1.6.2	When an Interrogator issues a command that uses delayed reply timing and the Tag successfully executes the command, after executing the command the Tag shall backscatter the reply shown in Table 6.13 and Figure 6.16, comprising a header (a 0-bit), the Tag's handle, and a CRC-16 calculated over the 0-bit and handle.	M	Tag	**By demonstration** Write the PC with value 0x3000. Verify that the Tag response is according Table 6.13 and Figure 6.16. T_5 should meet the limit in Table 6.16. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
92	6.3.1.6.2	The reply shall meet the T_5 limits in Table 6.16.	M	Tag	**By demonstration** Tested in compliance with Item 91.
93	6.3.1.6.2	When an Interrogator issues a command that uses delayed reply timing and the Tag encounters an error, the Tag shall backscatter an error code (see Annex I) during the CW period rather than the reply shown in Table 6.13	M	Tag	**By demonstration** Write the Stored-CRC with value 0x0000. Verify that the Tag respond with an error message. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
94	6.3.1.6.2	A Tag shall ignore Interrogator commands while processing a prior command that specified a delayed reply.	M	Tag	**By demonstration** Write the PC with value 0x3000. Issue a Req_RN command before the Tag has time to respond. Verify that the Tag ignores the Req_RN command and only processes the Write operation. (either with/without write error) Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: >=0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1 Note: For this test it shall be verified that the tag always has sufficient power available.
95	6.3.1.6.2	A delayed Tag reply shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. the Tag shall reply as if TRext=1 regardless of the TRext value in the Query that initiated the inventory round).	M	Tag	**By design**
96	6.3.1.6.3	An *in-process* reply may include multiple backscatter transmissions from Tag to Interrogator.	O	Interrogator	**By demonstration** Issue a valid command that employs an in-process reply. Verify that the Interrogator is able to decode all backscatter transmissions. The maximum number of backscatter transmissions shall be specified by the interrogator manufacturer. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
97	6.3.1.6.3	The first transmission shall meet the T_6 limits specified in Table 6.16; subsequent transmissions (if any) shall meet T_7.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
98	6.3.1.6.3	A Tag shall backscatter a transmission at least once every $T_{7(max)}$ while processing the command.	M	Tag	**By demonstration** Issue a valid command that employs an in-process reply. Verify that the Tag replies as shown in Table 6.14. T_6 and T_7 shall meet limits specified in Table 6.16. The number of backscatter transmissions shall be limited to MIN (Max. Number of backscatter transmissions, 4). Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
99	6.3.1.6.3	A Tag's *in-process* reply or replies shall be as shown in Table 6.14.	M	Tag	**By demonstration** Tested in compliance with Item 98.
100	6.3.1.6.3	The Tag replies shall be consistent for first and subsequent Tag transmissions – i.e. if the final reply includes length then all intermediate replies shall include length, and vice versa.	M	Tag	**By design**
101	6.3.1.6.3	While processing the command the Tag backscatters a transmission as shown in Table 6.14 at least once every $T_{7(max)}$. Done and header for these intermediate replies shall be zero, response shall be null, and if the replies include length then length=0000_h.	M	Tag	**By demonstration** Tested in compliance with Item 98.
102	6.3.1.6.3	All replies shall meet the T_6 and T_7 limits specified in Table 6.16. If the Interrogator observes a final reply with header=0 then the command completed successfully.	M	Tag	**By demonstration** Tested in compliance with Item 98.
103	6.3.1.6.3	All replies shall meet the T_6 and T_7 limits specified in Table 6.16. If the Interrogator observes a final reply with header=1 then the Tag encountered an error (see Annex I).	M	Tag	**By demonstration** Issue an command with an unsupported parameter that employs an in-process reply. Verify that the Tag replies an error (see Table 6.14 and Annex I). T_6 and T_7 shall meet limits specified in Table 6.16. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
104	6.3.1.6.3	Length shall comprise a 15-bit value field followed by an even parity bit (the number of 1's in the 16-bit length field shall be an even number).	M	Tag	**By design**
105	6.3.1.6.3	A Tag shall ignore Interrogator commands while processing a prior command that specified an *in-process* reply.	M	Tag	**By demonstration** Issue a valid command that employs an in-process reply. Issue a Req_RN command before the Tag has time to respond. Verify that the Tag ignores the Req_RN command and only processes the in-process reply. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: >=0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1 Note: For this test it shall be verified that the tag always has sufficient power available.
106	6.3.1.6.3	After issuing a command that uses an *in-process* reply an Interrogator shall transmit CW until the Interrogator either (a) observes a reply with done=1 indicating the Tag has finished executing the command, or (b) fails to observe a reply for at least $T_{6(max)}$ or $T_{7(max)}$ (as appropriate) indicating that the Tag failed to execute the command.	M	Interrogator	**By design**
107	6.3.1.6.3	An *in-process* Tag reply shall use the extended preamble shown in Figure 6.11 or Figure 6.15, as appropriate (i.e. the Tag shall reply as if TRext=1 regardless of the TRext value in the *Query* that initiated the inventory round).	M	Tag	**By design**
108	6.3.1.6.4	A Tag that implements a *Challenge* command, or any access command (other than *AuthComm*) that employs an *in-process* reply, shall implement a **C** flag and a ResponseBuffer with the following properties:	M	Tag	**By demonstration** Tested in compliance with Item 109.

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
109	6.3.1.6.4	A Tag shall set **C**=0 upon receiving either an access command with SenRep=0 (c.f. 6.3.2.12.3.10) or a *Challenge* command, and shall set **C**=1 after finishing its processing and storing its response (result or error code) in its ResponseBuffer.	TACC TACA TACF CE	Tag	**By demonstration** Read XPC_W1 value to verify that the initial value of C-flag is '0'. Issue a Challenge or an Authenticate command with SenRep=0. Read XPC_W1 value to verify that the value of C-flag has changed to '1'. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
110	6.3.1.6.4	The **C** flag shall be selectable using a *Select* command.	M	Tag	**By demonstration** Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
111	6.3.1.6.4	If an access command with SenRep=0 or a *Challenge* command specifies IncRepLen=0 then a Tag shall not include a length field with its stored response, so the first word of the stored response shall be at ResponseBuffer location 00_h.	TACC TACA TACF CE	Tag	**By demonstration** Issue a Challenge or an Authenticate command with SenRep=0 and IncRepLen=0. Verify that no length field is contained in the ResponseBuffer, by using the ReadBuffer command. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
112	6.3.1.6.4	If the command specifies IncRepLen=1 then ResponseBuffer bits 00_h – $0E_h$ shall contain the length of the stored response in bits, ResponseBuffer bit $0F_h$ shall contain an even parity bit that the Tag computes over bits 00_h – $0E_h$, and the first word of the stored response shall be at ResponseBuffer location 10_h. See Figure 6.17.	TACC TACA TACF CE	Tag	**By demonstration** Issue a Challenge or an Authenticate command with SenRep=0 and IncRepLen=1. Verify that the length field is contained in the ResponseBuffer, by using the ReadBuffer command. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
113	6.3.1.6.4	The maximum size of a stored response shall be 32 kbits.	TACC TACA TACF CE	Tag	**By design**
114	6.3.1.6.4	The maximum ResponseBuffer size shall be 32,784 bits (15 length bits, 1 parity bit, 32k response bits). A Tag manufacturer may limit the ResponseBuffer to a size less than this maximum.	TACC TACA TACF CE	Tag	**By design**
115	6.3.1.6.4	A Tag shall dynamically adjust its ResponseBuffer, on a command-by-command basis, to the required size.	TACC TACA TACF CE	Tag	**By design**
116	6.3.1.6.4	The ResponseBuffer shall be read-only to an Interrogator.	TACC TACA TACF CE	Tag	**By Design**
117	6.3.1.6.4	A Tag shall abort command processing and instead store an error code in its ResponseBuffer if and when it determines that response will overflow the ResponseBuffer (see Annex I).	TACC TACA TACF CE	Tag	**By Design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
118	6.3.1.6.4	A Tag shall retain data in its ResponseBuffer with the persistence of its **C** flag (see Table 6.20).	TACC TACA TACFC E	Tag	**By demonstration** Issue a Challenge or an Authenticate command with SenRep=0. Switch of the carrier after successful tag response. Read XPC_W1 after the persistence of the C flag. Verify that C=1. Issue a ReadBuffer command to verify that the Tag deallocated its ResponseBuffer. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
119	6.3.1.6.4	When **C** is 1 then a Tag shall maintain the data in its ResponseBuffer.	TACC TACA TACF CE	Tag	**By demonstration** Tested in compliance with Item 112.
120	6.3.1.6.4	When **C** is or becomes 0 then a Tag shall deallocate its ResponseBuffer.	TACC TACA TACF CE	Tag	**By demonstration** Tested in compliance with Item 119.
121	6.3.1.6.4	The maximum value for T_2 shall apply only to Tags in the **reply** or **acknowledged** states (see 6.3.2.6.3 and 6.3.2.6.4). For a Tag in the **reply** or **acknowledged** states, if T_2 expires (i.e. reaches its maximum value): • Without the Tag receiving a valid command, the Tag shall transition to the **arbitrate** state (see 6.3.2.6.2), • During the reception of a valid command, the Tag shall execute the command, • During the reception of an invalid command, the Tag shall transition to **arbitrate** upon determining that the command is invalid. In all other states the maximum value for T_2 shall be unrestricted. A Tag shall be allowed a tolerance of $20.0T_{pri} \leq T_{2(max)} \leq 32T_{pri}$ in determining whether T_2 has expired. “Invalid command” is defined in 6.3.2.12.	M	Tag	**By design**
122	6.3.1.6.4	T_1+T_3 shall not be less than T_4.	M	Tag	**By design**
123	6.3.2.1	Tag memory shall be logically separated into four distinct banks, each of which may comprise zero or more memory words.	M	Tag	**By demonstration** Tested in compliance with Item 132.
124	6.3.2.1	Reserved memory shall contain the kill and and/or access passwords, if passwords are implemented on the Tag.	M	Tag	**By demonstration** Tested in compliance with Item 132.
125	6.3.2.1	The kill password shall be stored at memory addresses 00_h to $1F_h$.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
126	6.3.2.1	The access password shall be stored at memory addresses 20_h to $3F_h$.	M	Tag	**By design**
127	6.3.2.1	EPC memory shall contain a CRC-16 at memory addresses 00_h to $0F_h$, Protocol-Control (PC) bits at memory addresses 10_h to $1F_h$, and a code (such as an EPC, and hereafter referred to as an EPC) that identifies the object to which the Tag is or will be attached beginning at address 20_h.	M	Tag	**By design**
128	6.3.2.1	TID memory shall contain an 8-bit ISO/IEC 15963 allocation class identifier at memory locations 00_h to 07_h. TID memory shall contain sufficient identifying information above 07_h for an Interrogator to uniquely identify the custom commands and/or optional features that a Tag supports.	M	Tag	**By demonstration** Singulate the Tag, read its TID memory, and verify the contents. Tag test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0
129	6.3.2.1	The logical addressing of all memory banks and User-memory files shall begin at zero (00_h).	M	Tag	**By design**
130	6.3.2.1	The backscatter shall fall on word boundaries (except in the case of a truncated reply – see 6.3.2.10.1.1).	M	Tag	**By design**
131	6.3.2.1	MemBank isshall be defined as follows: 00_2 Reserved 01_2 EPC 10_2 TID 11_2 User	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
132	6.3.2.1	Operations in one logical memory bank shall not access memory locations in another bank.	M	Tag	**By demonstration** "Check for memory overruns" Initialise the all (writable) memory locations in each bank with the value 0x0000. Issue a sequence of Write commands to write the PC with 0x3000 and the remaining EPC memory with 0x1111. Verify, using the Read command, that no memory location in another bank has been overwritten. Issue a sequence of Write commands to write the entire USER memory bank with 0x2222. Verify, using the Read command, that no memory location in another bank has been overwritten. Issue a sequence of Write commands to write the entire RESERVED memory bank with 0x3333. Verify, using the Read command, that no memory location in another bank has been overwritten. Issue a sequence of Write commands to attempt to write the entire TID memory bank with 0x4444. Verify, using the Read command, that no memory location in TID or another bank has been overwritten. <u>Test conditions:</u> Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
133	6.3.2.1	A *Write*, *BlockWrite*, or *BlockErase* shall not alter a Tag's **killed** status regardless of the memory address (whether valid or invalid) specified in the command.	M	Tag	**By demonstration** Tested in compliance with Item 132.
134	6.3.2.1	A Tag shall use the same block size for file allocation (see 6.3.2.11.3) as it does for the Block-Permalock command (see 6.3.2.12.3.9).	M	Tag	**By design**
135	6.3.2.1.1	If a Tag does not implement the kill and/or access password(s), the Tag shall logically operate as though it has zero-valued password(s) that are permanently read/write locked (see 6.3.2.10.3.5), and the corresponding physical memory locations in Reserved memory need not exist.	M	Tag	**By design**
136	6.3.2.1.1.1	The default (unprogrammed) value shall be zero.	M	Tag	**By design**
137		A Tag that does not implement a kill password shall behave as though it has a zero-valued kill password that is permanently read/write locked.	M	Tag	**By demonstration** **Add test method** Tested in compliance with Item 138.

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
138	6.3.2.1.1	A Tag shall not execute a password-based kill if its kill password is zero (see 6.3.2.12.3.4)	M	Tag	**By demonstration** Issue a *Kill* command to a Tag with a zero-valued kill password. Verify that the Tag backscatters an error code and does not execute the kill. Tag test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0
139	6.3.2.1.1.2	The default (unprogrammed) value shall be zero.	M	Tag	**By design**
140	6.3.2.1.1.2	A Tag that does not implement an access password shall behave as though it has a zero-valued access password that is permanently read/write locked	M	Tag	**By Demonstration** Attempt to write the access password. A Tag that does not implement an access password shall respond with an error message. A Tag that implements an access password shall respond with a valid response. Attempt to read the access password. A Tag that does not implement an access password shall respond with an error message. A Tag that implements an access password shall respond with a valid response. Tag test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0
141	6.3.2.1.2	The StoredCRC, StoredPC, EPC and XPC word(s) shall be stored MSB first (the EPC's MSB is stored in location 20_h).	M	Tag	**By design**
142	6.3.2.1.2.1	A Tag shall implement both a StoredCRC and a PacketCRC.	M	Tag	**By design**
143	6.3.2.1.2.1	A Tag shall compute and store its StoredCRC either (i) when an Interrogator writes or overwrites bits in the EPC (including in the StoredPC), or (ii) every time the Tag powers up.	M	Tag	**By design**
144	6.3.2.1.2.1	The Tag manufacturer shall choose whether the Tag implements (i) or (ii),	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
145	6.3.2.1.2.1	A Tag that implements case (i) shall perform its computing and storing as follows: The Tag first writes or overwrites the bits, then computes and stores a new StoredCRC, all within the reply times specified in Table 6.16 for the command (Write, BlockWrite, BlockErase, or Untraceable) that wrote or overwrote the bits. A Tag shall delay backscattering the success reply shown in Table 6.13 or Table 6.14 for the command that wrote or overwrote the bits until it has stored the new StoredCRC. The Tag shall store its StoredCRC in nonvolatile memory so that the StoredCRC persists through subsequent Tag power cycles. Alternatively, a Tag that implements case (ii) shall perform its computing and storing as follows: The Tag computes and stores the StoredCRC before the end of interval Ts or Ths (as appropriate) in Figure 6.3 or Figure 6.5, respectively.	M	Tag	**By design**
146	6.3.2.1.2.1	For both cases (i) and (ii) the Tag shall implement the StoredCRC by first calculating a CRC-16 (see 6.3.1.5) over the StoredPC and the EPC specified by the length (L) bits in the StoredPC, and then storing the thus-computed StoredCRC into EPC memory 00h to 0Fh, MSB first.	M	Tag	**By demonstration** Test for rewriteable Tags: Sequentially write a Tag's EPC, one 16-bit word at a time. Following each write, update the length field specified in the PC bits, power down the Tag, then power it up again and singulate it. Verify that the backscattered CRC-16 matches the backscattered EPC after each write operation. Test for prewritten Tags: Power up the Tag and singulate it. Verify that the backscattered CRC-16 matches the backscattered EPC. Tag test conditions for either case: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0
147	6.3.2.1.2.1	The Tag shall calculate the StoredCRC on word boundaries, shall deassert all Tag-computed StoredPC bit values (XI and UMI if Tag-computed) when performing the calculation, and shall omit XPC_W1 and XPC_W2 from the calculation.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
148	6.3.2.1.2.1	If an Interrogator attempts to write to EPC memory 00h – 0Fh then the Tag shall not execute the write and instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By demonstration** Issue a Write command on the specified memory location. Verify that the Tag responds with an error message. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
149	6.3.2.1.2.1	A Tag shall compute the PacketCRC as specified in 6.3.1.5 over the PC word, optional XPC word(s), and backscattered EPC, and shall send the PacketCRC MSB first.	M	Tag	**By design**
150	6.3.2.1.2.1	As required by 6.3.1.5 an Interrogator shall verify the integrity of the received PC word, optional XPC word or words, and EPC using the PacketCRC.	M	Interrogator	**By design**
151	6.3.2.1.2.2	A Tag shall implement a StoredPC in addresses 10h–1Fh of EPC memory. The bit assignments for this StoredPC shall be as shown in Table 6.18 and defined in Table 6.19.	M	Tag	**By design**
152	6.3.2.1.2.2	The StoredPC bits and values shall be as follows: Bits 10h – 14_h are written by an Interrogator and specify the length of the EPC that a Tag backscatters in response to an ACK, in words.	M	Interro-gator	**By design**
153	6.3.2.1.2.2	If a Tag only supports XI=0 then the maximum value for the EPC length field in the StoredPC shall be 11111_2 (allows a 496-bit EPC), as shown above. If a Tag supports XI=1 then the maximum value for the EPC length field in the StoredPC shall be 11101_2 (allows a 464-bit EPC).	M	Tag	**By design**
154	6.3.2.1.2.2	A Tag that supports XI=1 shall not execute a Write, BlockWrite, or Untraceable that attempts to write an EPC length field larger than 11101_2 and shall instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
155	6.3.2.1.2.2	Bit 15_h may be fixed by the Tag manufacturer or computed by the Tag. In the former (fixed) case, if the Tag does not have and is incapable of allocating memory to File_0 then the Tag manufacturer shall set bit 15_h to 0_2; if the Tag has or is capable of allocating memory to File_0 then the Tag manufacturer shall set bit 15_h to 1_2	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
156	6.3.2.1.2.2	In the latter (computed) case, both at power-up and upon writing the first word (bits 00_h – $0F_h$) of File_0 a Tag shall compute the logical OR of bits 03_h – 07h of File_0 and shall map the computed value into bit 15_h; if the Tag does not have memory allocated to File_0 then the logical OR result shall be 0_2.	O	Tag	**By design**
157	6.3.2.1.2.2	Regardless of the UMI method (fixed or computed), when an Interrogator writes the StoredPC the Tag shall not write and instead ignore the data value the Interrogator provides for bit 15_h.	O	Tag	**By design**
158	6.3.2.1.2.2	For a computed UMI, if an Interrogator deallocates File_0 (see 6.3.2.11.3) then the Tag shall set bit 15_h to 0_2 upon deallocation.	O	Tag	**By design**
159	6.3.2.1.2.2	Also for a computed UMI, the untraceability status of User memory (see 6.3.2.11.1) shall not change the UMI value (i.e. if UMI=1 when a Tag is traceable then UMI shall remain 1 even if an Interrogator instructs a Tag to untraceably hide User memory).	O	Tag	**By design**
160	6.3.2.1.2.2	If a Tag does not implement XPC_W1 then bit 16_h shall be fixed at 0_2 by the Tag manufacturer.	M	Tag	**By design**
161	6.3.2.1.2.2	If a Tag implements XPC_W1 then a Tag shall compute XI both at powerup and upon changing any bits of XPC_W1 (whether these bits are written or computed) and map the computed value into bit 16_h as follows: If T=0 then XI may be either (i) the logical OR of bits 210_h–217_h of EPC memory or (ii) the logical OR of bits 210_h–218_h of EPC memory; the Tag manufacturer shall choose whether the Tag implements (i) or (ii). If T=1 then XI is the logical OR of bits 210_h–$21F_h$ of EPC memory.	EAS TAC TACC TACA TACF CE O	Tag	**By design**
162	6.3.2.1.2.2	Regardless of whether XI is fixed or computed, when an Interrogator writes the StoredPC the Tag shall not write and instead ignore the data value the Interrogator provides for bit 16_h.	M	Tag	**By design**
163	6.3.2.1.2.2	If bit 17_h is 0_2, then the application is referred to as a GS1 EPCglobal™ Application and PC bits 18_h – $1F_h$ shall be as defined in this protocol. If bit 17_h is 1_2, then the application is referred to as a non-GS1 EPCglobal™ Application and PC bits 18_h – $1F_h$ shall be as defined in ISO/IEC 15961.	M	Tag	**By demonstration** Tag test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
164	6.3.2.1.2.2	**RFU or AFI (Reserved for Future Use or Application Family Identifier, bits 18_h – $1F_h$):** If T=0 then the Tag manufacturer (if the bits are not writeable) or an Interrogator (if the bits are writeable) shall set these bits to 00_h. If T=1 then the Tag manufacturer (if the bits are not writeable) or an Interrogator (if the bits are writeable) shall set these bits as specified in ISO/IEC 15961.	M	Tag	**By design**
165	6.3.2.1.2.2	If an Interrogator attempts to write L bit values that the Tag does not support then the Tag shall not execute the write operation and instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By design**
166	6.3.2.1.2.2	A Tag that supports XI=1 shall implement a PacketPC in addition to a StoredPC.	O	Tag	**By design**
167	6.3.2.1.2.2	Which PC word a Tag backscatters in reply to an ACK shall be as defined in Table 6.17.	M	Tag	**By design**
168	6.3.2.1.2.2	A PacketPC differs from a StoredPC in its **L** bits, which a Tag adjusts to match the length of the backscattered data that follow the PC word. Specifically, if **XI**=1 but **XEB**=0 then a Tag backscatters an XPC_W1 before the EPC, so the Tag shall add one to (i.e. increment) its L bits. If both **XI**=1 and **XEB**=1 then the Tag backscatters both an XPC_W1 and an XPC_W2 before the EPC, so the Tag shall add two to (i.e. double increments) its **L** bits. Because Tags that support XPC functionality have a maximum L value of 11101_2, double incrementing increases the value to 11111_2.	EAS TAC TACC TACA TACF CE O	Tag	**By design**
169	6.3.2.1.2.2	A Tag shall not, under any circumstances, allow its L bits to roll over to 00000_2.	M	Tag	**By design**
170	6.3.2.1.2.2	If a Tag has **T**=0, **XI**=0, implements an XPC_W1, and is not truncating then the Tag substitutes the 8 LSBs of XPC_W1 (i.e. EPC memory 218_h – $21F_h$) for the 8 LSBs of the StoredPC (i.e. PC memory 18_h – $1F_h$) in its reply. Because a Tag calculates its PacketCRC over the backscattered data bits (see 6.3.2.1.2.1), when the Tag does this substitution then it shall calculate its PacketCRC over the 8 substituted PC_W1 LSBs rather than over the 8 StoredPC LSBs.	EAS TAC TACC TACA TACF CE O	Tag	**By design**
171	6.3.2.1.2.2	An Interrogator shall support Tag replies with **XI**=0, **XI** =1, or both **XI**=1 and **XEB**=1.	M	Interrogator	**By design**
172	6.3.2.1.2.2	If a Tag has a response (result or error code) in its ResponseBuffer (i.e. **C**=1) and the Interrogator set immed=1 in the Challenge command that preceded the inventory round then a Tag shall concatenate response and a CRC-16 calculated over response to its reply to an ACK (see Table 6.17).	O	Tag	**By design**
173	6.3.2.1.2.3	The EPC for a GS1 EPCglobal™ Application shall be as defined in the GS1 EPC™ Tag Data Standard.	O	Tag	**By design**
174	6.3.2.1.2.4	The EPC for a non-EPCglobal™ Application shall be as defined in ISO/IEC 15961.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
175	6.3.2.1.3	TID memory locations 00_h to 07_h shall contain either $E0_h$ or $E2_h$. ISO/IEC 15963 class-identifier value —	M	Tag	**By design**
176	6.3.2.1.3	TID memory locations above 07_h shall be defined according to the registration authority defined by this class-identifier value and shall contain, at a minimum, sufficient information for an Interrogator to uniquely identify the custom commands and/or optional features that a Tag supports.	O	Tag	**By design**
177	6.3.2.1.3	If the class identifier is $E2_h$ then TID memory above 07_h shall be configured as follows: • 08_h: XTID (**X**) indicator (whether a Tag implements an XTID – see 5.2) • 09_h: Security (**S**) indicator (whether a Tag supports the *Authenticate* and/or *Challenge* commands) • $0A_h$: File (**F**) indicator (whether a Tag supports the *FileOpen* command) • $0B_h$ to 13_h: A 9-bit Tag mask-designer identifier (obtainable from the registration authority) • 14_h to $1F_h$: A Tag-manufacturer-defined 12-bit Tag model number • Above $1F_h$: As defined in the GS1 EPC Tag Data Standard	O	Tag	**By design**
178	6.3.2.1.4	If File_0 of User memory exists and has not yet been written then the 5 LSBs of the first byte (i.e. File_0 memory addresses 03_h to 07_h) shall have the default value 00000_2.	O	Tag	**By design**
179	6.3.2.1.4.1	If a Tag implements User memory then the file encoding shall be as defined in the GS1 EPC Tag Data Standard.	O	Tag	**By design**
180	6.3.2.1.4.2	If a Tag implements User memory then the file encoding shall be as defined in ISO/IEC 15961 and 15962.	O	Tag	**By design**
181	6.3.2.2	Interrogators shall support and Tags shall provide 4 sessions (denoted S0, S1, S2, and S3).	M	Tag and Interro-gator	**By design**
182	6.3.2.2	Tags shall participate in one and only one session during an inventory round.	M	Tag	**By design**
183	6.3.2.2	A Tag shall maintain an independent inventoried flag for each of its four sessions.	M	Tag	**By design**
184	6.3.2.2	Tags participating in an inventory round in one session shall neither use nor modify the **inventoried** flag for a different session.	M	Tag	**By design**
185	6.3.2.2	A Tag's **inventoried** flags shall have the set and persistence times shown in Table 6.20.	M	Tag	**By design**
186	6.3.2.2	A Tag shall power-up with its **inventoried** flags set as follows: the S0 **inventoried** flag shall be set to *A*.	M	Tag	**By design** Tested in compliance with 6.3.2.3, Table 6.20

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
187	6.3.2.2	A Tag shall power-up with its **inventoried** flags set as follows: the S1 **inventoried** flag shall be set to either *A* or *B*, depending on its stored value, unless the flag was set longer in the past than its persistence time, in which case the Tag shall power-up with its S1 **inventoried** flag set to *A*.	M	Tag	**By design**
188	6.3.2.2	A Tag shall power-up with its **inventoried** flags set as follows: the S2 **inventoried** flag shall be set to either *A* or *B*, depending on its stored value, unless the Tag has lost power for a time greater than its persistence time, in which case the Tag shall power-up with the S2 **inventoried** flag set to *A*.	M	Tag	**By design**
189	6.3.2.2	A Tag shall power-up with its **inventoried** flags set as follows: the S3 **inventoried** flag shall be set to either *A* or *B*, depending on its stored value, unless the Tag has lost power for a time greater than its persistence time, in which case the Tag shall power-up with its S3 **inventoried** flag set to *A*.	M	Tag	**By design**
190	6.3.2.2	Tag shall refresh its S2 and S3 flags while powered, meaning that every time a Tag loses power its S2 and S3 **inventoried** flags shall have the persistence times shown in Table 6.20.	M	Tag	**By design**
191	6.3.2.2	A Tag shall not change the value of its S1 **inventoried** flag from *B* to *A*, as the result of a persistence timeout, while the Tag is participating in an inventory round, is in the midst of being inventoried, or is in the midst of being accessed.	M	Tag	**By design**
192	6.3.2.2	If a Tag's S1 flag persistence time expires during an inventory round then the Tag shall change the flag to *A* only (i) as instructed by an Interrogator (e.g. by a *QueryAdjust* or *QueryRep* with matching session at the end of an inventory or access operation), or (ii) at the end of the round (e.g. upon receiving a *Select* or *Query*). In case (i), if the Tag's S1 flag persistence time expires while the Tag is in the midst of being inventoried or accessed then the Tag shall change the flag to *A* at the end of the inventory or access operation. In case (ii), the Tag shall invert its S1 flag prior to evaluating the *Select* or *Query*.	M	Tag	**By design**
193	6.3.2.3	Tags shall implement a selected flag, **SL**, which an Interrogator may assert or deassert using a *Select* command.	M	Tag	**By design**
194	6.3.2.3	A Tag's **SL** flag shall have the set and persistence times shown in Table 6.20.	M	Tag	**By design** Tested in compliance with 6.3.2.3, Table 6.20
195	6.3.2.3	A Tag shall power-up with its **SL** flag either asserted or deasserted, depending on the stored value, unless the Tag has lost power for a time greater than the **SL** persistence time, in which case the Tag shall power-up with its **SL** flag deasserted (set to ~**SL**).	M	Tag	**By design**
196	6.3.2.3	A Tag shall refresh its **SL** flag when powered, meaning that every time a Tag loses power its **SL** flag shall have the persistence times shown in Table 6.20.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
197	6.3.2.3, Table 6.20	For a randomly chosen and sufficiently large Tag population, 95% of the Tag persistence times shall meet the persistence requirement, with a 90% confidence interval.	M	Tag	**By design** Tag manufacturers shall provide data and analysis demonstrating that Tags meet the persistence requirements of Table 6.20.
198	6.3.2.4	A Tag's **C** flag (see 6.3.2.1.2.5) shall have the set and persistence times shown in Table 6.20.	O	Tag	**By design** Tested in compliance with 6.3.2.3, Table 6.20
199	6.3.2.4	A Tag shall refresh its **C** flag when powered, meaning that every time a Tag loses power its **C** flag shall have the persistence shown in Table 6.20 (of course, if a Tag has a zero-second persistence time then even if the Tag powers down momentarily its **C** flag will be deasserted).	C	Tag	**By design**
200	6.3.2.5	If a Tag implements a security timeout then it shall use a single timeout timer, so a security timeout caused by one command failure (such as a failed *Challenge*) shall cause a Tag to disallow all commands for which the Tag implements a security timeout until the end of the timeout period.	EAS TAC TACC TACA TACF CE O	Tag	**By design**
201	6.3.2.6	Tags shall implement the states and the slot counter shown in Figure 6.21.	M	Tag	**By demonstration** Tag test: Tag manufacturers shall supply a population of Tags for testing. The testing laboratory shall exercise all of the states and state transitions shown in Figure 6.21 by selecting, singulating, inventorying, reading, writing, accessing, and (for Tags that implement kill) killing the Tags. Tag test conditions: Temp: 23 +/– 3 ºC Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0
202	6.3.2.6.1	Tags shall implement a **ready** state.	M	Tag	**By design** – inherently tested with other tests Tested in compliance with 6.3.2.6, Figure 6.21
203	6.3.2.6.1	Upon entering an energizing RF field a Tag that is not **killed** shall enter **ready**.	M	Tag	**By design**
204	6.3.2.6.1	The Tag shall remain in **ready** until it receives a *Query* command (see 6.3.2.12.2.1) whose **inventoried** parameter (for the session specified in the *Query*) and sel parameter match its current flag values.	M	Tag	**By design**
205	6.3.2.6.1	Matching Tags shall draw a *Q*-bit number from their RNG (see 6.3.2.7), load this number into their slot counter, and transition to the **arbitrate** state if the number is nonzero, or to the **reply** state if the number is zero.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
206	6.3.2.6.1	If a Tag in any state except **killed** loses power it shall return to **ready** upon regaining power.	M	Tag	**By design**
207	6.3.2.6.2	Tags shall implement an **arbitrate** state.	M	Tag	**By design** Tested in compliance with 6.3.2.6, Figure 6.21
208	6.3.2.6.2	A Tag in **arbitrate** shall decrement its slot counter every time it receives a *QueryRep* command (see 6.3.2.12.2.3) whose session parameter matches the session for the inventory round currently in progress, and it shall transition to the **reply** state and backscatter an RN16 when its slot counter reaches 0000_h.	M	Tag	**By design**
209	6.3.2.6.2	Tags that return to **arbitrate** (for example, from the **reply** state) with a slot value of 0000_h shall decrement their slot counter from 0000_h to $7FFF_h$ at the next *QueryRep* (with matching session) and, because their slot value is now nonzero, shall remain in **arbitrate**.	M	Tag	**By design**
210	6.3.2.6.3	Tags shall implement a **reply** state.	M	Tag	Tested in compliance with 6.3.2.6, Figure 6.21
211	6.3.2.6.3	Upon entering **reply** a Tag shall backscatter an RN16.	M	Tag	**By design**
212	6.3.2.6.3	If the Tag receives a valid acknowledgement (*ACK*) it shall transition to the **acknowledged** state, backscattering the reply shown in Table 6.17.	M	Tag	**By design**
213	6.3.2.6.3	If the Tag fails to receive an *ACK* within time $T_{2(max)}$, or receives an invalid *ACK* or an *ACK* with an erroneous RN16, it shall return to **arbitrate**.	M	Tag	**By design**
214	6.3.2.6.3	Tag and Interrogator shall meet all timing requirements specified in Table 6.16.	M	Tag and Interrogator	**By design** Tested in compliance with 6.3.1.6, Table 6.16
215	6.3.2.6.4	Tags shall implement an **acknowledged** state.	M	Tag	Tested in compliance with 6.3.2.6, Figure 6.21
216	6.3.2.6.4	If a Tag in the **acknowledged** state receives a valid *ACK* containing the correct RN16 then it shall re-backscatter the reply shown in Table 6.17.	M	Tag	**By design**
217	6.3.2.6.4	If a Tag in the **acknowledged** state fails to receive a valid command within time $T_{2(max)}$ it shall return to **arbitrate**.	M	Tag	**By design**
218	6.3.2.6.4	Tag and Interrogator shall meet all timing requirements specified in Table 6.16.	M	Tag and Interrogator	**By design** Tested in compliance with 6.3.1.6, Table 6.16
219	6.3.2.6.5	Tags shall implement an **open** state.	M	Tag	Tested in compliance with 6.3.2.6, Figure 6.21
220	6.3.2.6.5	A Tag in the **acknowledged** state whose access password is nonzero shall transition to **open** upon receiving a *Req_RN* command, backscattering a new RN16 (denoted handle) that the Interrogator shall use in subsequent commands and the Tag shall use in subsequent replies.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
221	6.3.2.6.5	If a Tag in the **open** state receives a valid *ACK* containing the correct handle then it shall re-backscatter the reply shown in Table 6.17.	M	Tag	**By design**
222	6.3.2.6.5	Tag and Interrogator shall meet all timing requirements specified in Table 6.16 except $T_{2(max)}$; in the **open** state the maximum delay between Tag response and Interrogator transmission is unrestricted.	M	Tag and Interrogator	**By design** Tested in compliance with 6.3.1.6, Table 6.16
223	6.3.2.6.6	Tags shall implement a **secured** state.	M	Tag	**By design** Tested in compliance with 6.3.2.6, Figure 6.21
224	6.3.2.6.6	A Tag in the **acknowledged** state whose access password is zero shall transition to **secured** upon receiving a *Req_RN* command, backscattering a new RN16 (denoted handle) that the Interrogator shall use in subsequent commands and the Tag shall use in subsequent replies.	M	Tag and Interro-gator	**By design**
225	6.3.2.6.6	A Tag in the **open** state shall transition to **secured** following a successful *Access* command sequence or Interrogator authentication (where success in the latter case is defined by the cryptographic suite specified in the *Authenticate* command that initiated the authentication), maintaining the same handle that it previously backscattered when it transitioned from the **acknowledged** to the **open** state.	M	Tag	**By design**
226	6.3.2.6.6	If a Tag in the **secured** state receives a valid *ACK* containing the correct handle then it shall re-backscatter the reply shown in Table 6.17.	M	Tag	**By design**
227	6.3.2.6.6	Tag and Interrogator shall meet all timing requirements specified in Table 6.16 except $T_{2(max)}$; in the **secured** state the maximum delay between Tag response and Interrogator transmission is unrestricted.	M	Tag and Interrogator	**By design** Tested in compliance with 6.3.1.6, Table 6.16
228	6.3.2.6.7	Tags shall implement a **killed** state.	M	Tag	**By design** Tested in compliance with 6.3.2.6, Figure 6.21
229	6.3.2.6.7	A Tag in either the **open** or **secured** states shall enter the **killed** state upon receiving a successful password-based *Kill*-command sequence with a correct nonzero kill password and handle.	M	Tag	**By design**
230	6.3.2.6.7	A Tag in the **secured** states shall enter the **killed** state upon a successful authenticated *Kill* (see 6.3.2.12.3.4).	TACC, TACA109	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
231	6.3.2.6.7	Upon entering the **killed** state a Tag shall notify the Interrogator that the kill operation was successful, and shall not respond to an Interrogator thereafter.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
232	6.3.2.6.7	**Killed** Tags shall remain in the **killed** state under all circumstances, and shall immediately enter **killed** upon subsequent power-ups.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
233	6.3.2.6.8	Tags shall implement a 15-bit slot counter.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
234	6.3.2.6.8	Upon receiving a *Query* or *QueryAdjust* command a Tag shall preload into its slot counter a value between 0 and 2^Q-1, drawn from the Tag's RNG (see 6.3.2.7).	M	Tag	**By design**
235	6.3.2.6.8	Tags whose slot counter reached 0000_h, who replied, and who were not **acknowledged** (including Tags that responded to an original *Query* and were not **acknowledged**) shall return to **arbitrate** with a slot value of 0000_h and shall decrement this slot value from 0000_h to $7FFF_h$ at the next *QueryRep*.	M	Tag	**By demonstration** Test in compliance with 6.3.2.6, Figure 6.21
236	6.3.2.6.8	The slot counter shall be capable of continuous counting, meaning that, after the slot counter rolls over to $7FFF_h$ it begins counting down again, thereby effectively preventing subsequent replies until the Tag loads a new random value into its slot counter.	M	Tag	**By demonstration** Test in compliance with 6.3.2.6, Figure 6.21
237	6.3.2.7	Tags shall implement a random or pseudo-random number generator (RNG).	M	Tag	**By design**
238	6.3.2.7	The RNG shall meet the following randomness criteria independent of the strength of the energizing RF field, the R=>T link rate, and the data stored in the Tag (including but not limited to the StoredPC, XPC word or words, EPC, and StoredCRC).	M	Tag	**By design**
239	6.3.2.7	Tags shall generate 16-bit random or pseudo-random numbers (RN16) using the RNG, and shall have the ability to extract *Q*-bit subsets from its RN16 to preload the Tag's slot counter (see 6.3.2.6.8).	M	Tag	**By design**
240	6.3.2.7	Tags shall have the ability to temporarily store at least two RN16s while powered, to use, for example, as a handle and a 16-bit cover-code during password transactions (see Figure 6.24 or Figure 6.26).	M	Tag	**By design**
241	6.3.2.7	The probability that any RN16 drawn from the RNG has value RN16 = *j*, for any *j*, shall be bounded by $0.8/2^{16} < P(RN16 = j) < 1.25/2^{16}$.	M	Tag	**By design** Tag manufacturers shall provide data and analysis demonstrating that Tags meet the requirements of 6.3.2.7.
242	6.3.2.7	For a Tag population of up to 10,000 Tags, the probability that any two or more Tags simultaneously generate the same sequence of RN16s shall be less than 0.1%, regardless of when the Tags are energized.	M	Tag	**By design** Tag manufacturers shall provide data and analysis demonstrating that Tags meet the requirements of 6.3.2.7.
243	6.3.2.7	An RN16 drawn from a Tag's RNG 10ms after the end of T_r in Figure 6.3 shall not be predictable with a probability greater than 0.025% if the outcomes of prior draws from the RNG, performed under identical conditions, are known.	M	Tag	**By design** Tag manufacturers shall provide data and analysis demonstrating that Tags meet the requirements of 6.3.2.7.
244	6.3.2.11	A Tag shall execute access commands only in the states shown in Table 6.27.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
245	6.3.2.11	A Tag shall treat as invalid (see Table C.30) optional access commands that it does not support.	M	Tag	**By design**
246	6.3.2.11.1	A Tag, once killed, shall not respond to an Interrogator thereafter.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
247	6.3.2.11.1	If the Interrogator issues a command with new data or half-password, then it shall first issue a *Req_RN* to obtain a new RN16 and shall use this RN16 for the cover-coding.	M	Interro-gator	**By design**
248	6.3.2.11.1	Neither Tag nor Interrogator shall consider themselves authenticated following an Access-command sequence.	M	Tag and Interro-gator	**By design**
249	6.3.2.11.1	A Tag or an Interrogator shall only consider themselves authenticated after executing a cryptographic authentication in accordance with a cryptographic suite.	M	Tag and Interro-gator	**By design**
250	6.3.2.11.2	Table 6.28 shows which commands an Interrogator may, and an authenticated Interrogator shallencapsulate in an *AuthComm*.	M	Interro-gator	**By design**
251	6.3.2.11.2	Table 6.28 shows which commands an Interrogator may, and an authenticated Interrogator shall, encapsulate in a *SecureComm*.	M	Interro-gator	**By design**
252	6.3.2.11.2	The Tag manufacturer shall choose the number and type of cryptographic suites that a Tag supports; this assignment shall not be alterable in the field. An Interrogator selects one from among the implemented cryptographic suites using the CSI field in the *Challenge* and *Authenticate* commands.	M	Tag and Interro-gator	**By design**
253	6.3.2.11.2	The Tag manufacturer shall choose the number of available keys and assign them to the cryptographic suite(s); this assignment shall not be alterable in the field.	M	Tag	**By design**
254	6.3.2.11.2	No two keys shall have the same number, even if used for different cryptographic suites.	M	Tag	**By design**
255	6.3.2.11.2	A Tag shall not indicate where in memory it stores its keys, nor shall it allow an Interrogator to read this memory location.	M	Tag	**By design**
256	6.3.2.11.2	A Tag that supports the *Untraceable* command shall provide the Tag privileges shown in Table 6.22.	O	Tag	**By design**
257		A Tag that supports one or more cryptographic suites shall provide the Tag privileges shown in Table 6.23.	O	Tag	**By design**
258	6.3.2.11.2	A Tag that implements the *TagPrivilege* command shall permit an Interrogator that authenticated itself as a crypto superuser in a cryptographic suite to: • change the value of any key in that cryptographic suite, including its own, using a *KeyUpdate*. • read or modify privileges (value in Table 6.23) for any key in that cryptographic suite, including its own.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
259	6.3.2.11.2	A Tag shall not permit an Interrogator that did not authenticate itself as a crypto superuser to: • change the value of any key other than the one it used to authenticate itself. • read or modify privileges (value in Table 6.23) for any key other than the one it used to authenticate itself • assert a deasserted privilege (value in Table 6.23) for the key it used to authenticate itself.	O	Tag	**By design**
260	6.3.2.11.2	A Tag that supports the *TagPrivilege* command shall permit an Interrogator that supplies the access password (even if zero-valued) or a key to deassert a privilege for the access password or that key, respectively, regardless of the CryptoSuperuser value.	O	Tag	**By design**
261	6.3.2.11.2	A Tag that receives a *TagPrivilege* that attempts to change an unchangeable Tag privilege value shall not execute the *TagPrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
262	6.3.2.11.2	After a successful Interrogator authentication a Tag in the **open** state shall transition to the **secured** state.	M	Tag	**By design**
263	6.3.2.11.2	The authenticated Interrogator shall subsequently encapsulate all commands designated "Mandatory Encapsulation" in Table 6.28 in an *AuthComm* or *SecureComm*.	M	Tag	**By design**
264	6.3.2.11.2	If a Tag receives such a command from an authenticated Interrogator without encapsulation then it shall not execute the command and instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By design**
265	6.3.2.11.2	A Tag shall transition back to the **open** state, reset its cryptographic engine, and revert to **open**-state file privileges (see below) when an authenticated Interrogator loses its authentication.	M	Tag	**By design**
266	6.3.2.11.2	If the Tag was not previously authenticated by a *Challenge* or *Authenticate* command then it shall not execute the command and instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By design**
267	6.3.2.11.2	If a condition of a cryptographic suite causes a Tag to transition from the **open** or **secured** state to the **arbitrate** state then the Tag (i) shall not change the value of its inventoried flag, and (ii) shall reset its cryptographic engine.	M	Tag	**By design**
268	6.3.2.11.3	A Tag that supports File_N, N>0 shall implement *FileOpen*; it may implement *FileList*, *FileSetup, and FilePrivilege* as well.	O	Tag	**By design**
269	6.3.2.11.3	If a Tag implements a single file then that file shall be File_0.	O	Tag	**By design**
270	6.3.2.11.3	A Tag with User memory shall open File_0 upon first entering the **open** or **secured** state.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
271	6.3.2.11.3	A Tag shall have only a single file open at any time. All access commands operate on the currently open file.	O	Tag	**By design**
272	6.3.2.11.3	Each file shall have an 8-bit FileType and a 10-bit FileNum, unless a Tag does not support any file-management access commands, in which case a Tag that implements File_0 may omit FileType and FileNum.	O	Tag	**By design**
273	6.3.2.11.3	A Tag manufacturer shall preassign a FileType to each file supported by the Tag.	O	Tag	**By design**
274	6.3.2.11.3	A Tag manufacturer shall preassign a unique FileNum to each file supported by the Tag.	O	Tag	**By design**
275	6.3.2.11.3	The files may have different size (including zero size). FileNum=0000000000_2 shall be reserved for the base file (File_0) of User memory	O	Tag	**By design**
276	6.3.2.11.3	FileNum=1111111111_2 shall be RFU. This protocol recommends, but does not require, that Tag manufacturers number files sequentially.	M	Tag	**By design**
277	6.3.2.11.3	Upon receiving a *FileOpen* a Tag shall first close the currently open file and then open the new file, with the new file's starting address mapped to 00_h of User memory.	O	Tag	**By design**
278	6.3.2.11.3	A Tag manufacturer shall precreate all files; the number of files shall not be changeable in the field.	O	Tag	**By design**
279	6.3.2.11.3	**Static:** A manufacturer of a *static* Tag shall preallocate all User memory to files.	O	Tag	**By design**
280	6.3.2.11.3	A Tag manufacturer shall decide where a Tag stores its FileType and FileNum data and may choose a readable portion of memory (if desired). Regardless of the location, a Tag shall not allow an Interrogator to modify a file's type by any command except *FileSetup*, and shall not allow an Interrogator to modify a FileNum by any means.	O	Tag	**By design**
281	6.3.2.11.3	Files may range in size from a minimum of zero to a maximum of 1022 blocks. Commands that include a FileSize parameter use 10 bits to specify sizes from zero to 1022 blocks (0000000000_2 – 1111111110_2, respectively). FileSize 1111111111_2 shall be RFU.	O	Tag	**By design**
282	6.3.2.11.3	A Tag manufacturer shall predefine a single fixed, unchangeable block size that the Tag shall use for all file allocation as well as for the *BlockPermalock* command.	O	Tag	**By design**
283	6.3.2.11.3	Tag manufacturers shall not use block sizes exceeding 1024 words. Tag replies that return a BlockSize value use 10 bits to specify the size from one (0000000000_2) to 1024 (1111111111_2) words. BlockSize does not have an RFU value.	O	Tag	**By design**
284	6.3.2.11.3	If a Tag supports File_0 then it shall provide the file privileges shown in Table 6.24.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
285	6.3.2.11.3	If a Tag supports File_N, N>0 then it shall also provide the file privileges shown in Table 6.25.	O	Tag	**By design**
286	6.3.2.11.3	A Tag with M files shall implement M 4-bit **open**-state file privileges, one for each file.	O	Tag	**By design**
287	6.3.2.11.3	A Tag with M files shall implement M 4-bit **secured**-state access-password file privileges, one for each file.	O	Tag	**By design**
288	6.3.2.11.3	. A Tag with M files and N keys shall implement M×N 4-bit **secured**-state key file privileges.	O	Tag	**By design**
289	6.3.2.11.3	In this latter case, if an Interrogator sends a 4-bit privilege with either MSB being nonzero then the Tag shall not execute the *FilePrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
290	6.3.2.11.3	A Tag shall permit an Interrogator that accessed or authenticated itself as a file superuser to: • read or assign a new 4-bit privilege for the **open** state, access password, or any key (including its own) regardless of the cryptographic suite to which the key is assigned, for the currently open file, using a *FilePrivilege* command. • change the FileType of the currently open file using a *FileSetup* command, for a *static* or a *dynamic* Tag. • resize the currently open file using a *FileSetup* command, but only if the file contains no permalocked or permaunlocked memory and only if the Tag is *dynamic*.	O	Tag	**By design**
291	6.3.2.11.3	A Tag shall not permit an Interrogator that did not access or authenticate itself as a file superuser to: • read or assign the 4-bit privilege for the **open** state, for the currently open file. • read or assign the 4-bit privilege for the access password or for any key other than the one it used to enter the **secured** state, for the currently open file. • increase the privileges (move down one or more rows in Table 6.24 or Table 6.25) for the access password or for any key, for the currently open file.	O	Tag	**By design**
292	6.3.2.11.3	If the access password or key that a Tag used to enter the **secured** state has DecFilePriv=1 (see Table 6.22 and Table 6.23) then a Tag shall permit an Interrogator to self-reduce its privileges (move up one or more rows in Table 6.24 or Table 6.25) to the currently open file for this access password or key.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
293	6.3.2.11.3	If *Read* is "×" for a privilege value then a Tag shall behave as if the memory location does not exist.	O	Tag	**By design**
294	6.3.2.11.3	Otherwise, if *Write*, *BlockWrite*, or *BlockErase* are "×" then the Tag shall behave as if the memory location is permalocked; and if *Lock* or *BlockPermalock* are "×" then the Tag shall behave as if the memory location is neither lockable nor unlockable.	O	Tag	**By design**
295	6.3.2.11.3	If *FilePrivilege* or *FileSetup* are "×" then the Tag shall behave as if the Interrogator has insufficient privileges.	O	Tag	**By design**
296	6.3.2.11.3	If a Tag implements the *BlockPermalock* command then all files shall support the *BlockPermalock* command.	O	Tag	**By design**
297	6.3.2.11.3	If a Tag's User memory is untraceably hidden then the Tag shall only execute a *FileOpen*, *FileList*, *FileSetup*, or *FilePrivilege* issued by an Interrogator with an asserted Untraceable privilege (see Table 6.22 and Table 6.23); if the Interrogator has a deasserted Untraceable privilege then the Tag shall treat these commands' parameters as unsupported (see Table C.30).	O	Tag	**By design**
298	6.3.2.11.3	A Tag shall not permit a permalocked portion of memory to be erased or overwritten, except for the **L** and **U** bits in EPC memory, which an Interrogator with an asserted Untraceable privilege may overwrite.	M	Tag	**By design**
299	6.3.2.11.3	In some instances a *dynamic* Tag may allow file resizing. Whether a Tag allows resizing shall depend on whether the Tag accepts a *FileSetup* command (varies by privilege and state), whether the Tag has free memory available for the resizing, and whether the file or any blocks in it are permalocked or permaunlocked.	O	Tag	**By design**
300	6.3.2.12	Interrogator-to-Tag commands shall use the command codes, protection, and parameters shown in Table 6.28.	M	Interro-gator	**By design**
301	6.3.2.12	*QueryRep*, *ACK*, *Query*, *QueryAdjust,* and *NAK* have the unique command lengths shown in Table 6.28. No other commands shall have these lengths.	M	Interro-gator	**By design**
302	6.3.2.12	If a Tag receives one of these commands with an incorrect length then it shall treat the command as invalid (see Table C.30)..	M	Tag	**By design**
303	6.3.2.12	An authenticated *Kill* shall be encapsulated in a *SecureComm* or an *AuthComm*; a password-based *Kill* shall not be encapsulated.	M	Interrogat or	**By design**
304	6.3.2.12.1.1	Interrogators and Tags shall implement the *Select* command shown in Table 6.29.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
305	6.3.2.12.1.1	A *Select* that modifies the **SL** flag shall not modify an **inventoried** flag, and vice versa.	M	Tag	**By design**
306	6.3.2.12.1.1	A Tag shall ignore a *Select* whose Target is 101_2, 110_2, or 111_2.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
307	6.3.2.12.1.1	If MemBank=00_2 then an Interrogator shall set Pointer to 00_h;	M	Interrogator	**By design**
308	6.3.2.12.1.1	if a Tag receives a *Select* with MemBank=00_2 and a nonzero Pointer value then it shall ignore the *Select*.	M	Tag	**By design**
309	6.3.2.12.1.1	If Mem-Bank=00_2 then an Interrogator shall set Length=00001000_2;	M	Interrogator	**By design**
310	6.3.2.12.1.1	if a Tag receives a *Select* with MemBank=00_2 and Length<>00001000_2 then it shall ignore the *Select*.	M	Tag	**By design**
311	6.3.2.12.1.1	An untraceable Tag shall process a *Select* with MemBank=002 whose User memory is traceable, or with MemBank<>002 whose Mask operates on a completely traceable bit string.	M	Tag	**By design**
312	6.3.2.12.1.1	A Tag shall treat as not-matching a *Select* command whose Mask includes untraceably hidden memory.	M	Tag	**By design**
313	6.3.2.12.1.1	Truncate indicates whether a Tag's backscattered reply shall be truncated to those EPC bits that follow Mask.	M	Tag	**By design**
314	6.3.2.12.1.1	If an Interrogator asserts Truncate, and if a subsequent *Query* specifies Sel=10 or Sel=11, then a matching Tag shall truncate its *ACK* reply to the portion of the EPC immediately following Mask, followed by a PacketCRC.	M	Tag	**By design**
315	6.3.2.12.1.1	If an Interrogator asserts Truncate, it shall assert it in the last *Select* that the Interrogator issues prior to sending a *Query*, only if the *Select* has Target = 100_2, and only if Mask ends in the EPC.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
316	6.3.2.12.1.1	Tags shall decide whether to truncate its backscattered EPC on the basis of the most recently received valid *Select* (i.e. not ignored and matching or not-matching).	M	Tag	**By demonstration:** Issue a *Select* command and a *Query* command to a Tag with a 96 bit EPC and the EPC starting with 001 and the parameters as follows: *Select* Command = 1010 Target = 100 Action = 000 MemBank = 01 Pointer = 0010 0000 Length = 0000 0011 Mask = 001 Truncate = 1 *Query* Command = 1000 DR = 0 M = 00 TRext = 0 Sel = 11 Session = 00 Target = 0 Q = 0000 Verify that the tag replies with a truncated EPC. Issue a *Select* command and a *Query* command to a Tag with a 96 bit EPC and the EPC starting with 001 and the parameters as follows: *Select* Command = 1010 Target = 100 Action = 100 MemBank = 01 Pointer = 0010 0000 Length = 0000 0011 Mask = 000 Truncate = 1 *Query* Command = 1000 DR = 0 M = 00 TRext = 0 Sel = 11 Session = 00 Target = 0 Q = 0000 Verify that the tag replies with a non-truncated EPC. Tag test conditions: Temp: 23 +/– 3 °C Freq: 860 & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK PW: 0.5 Tari Modulation depth: 90% Rise/fall time: ≤ 0.33 Tari Tari: 25 µs RTcal: 75 µs TRcal: 100 µs DR: 8 M: 1 TRext: 0
317	6.3.2.12.1.1	If a Tag receives a *Select* with Truncate=1 and Target<>100_2 or MemBank<>01_2 the Tag shall ignore the *Select*.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
318	6.3.2.12.1.1	If a Tag receives a *Select* in which Truncate=1, MemBank=01_2, but Mask ends outside the EPC specified by the L bits in the StoredPC then the Tag shall be not matching.	M	Tag	**By design**
319	6.3.2.12.1.1	A Tag shall preface its truncated reply with five leading zeros (00000_2) inserted between the preamble and the truncated reply.	M	Tag	**By design**
320	6.3.2.12.1.1	A Tag shall power-up with Truncate=0.	M	Tag	**By design**
321	6.3.2.12.1.1	Mask may end at the last bit of the EPC, in which case a truncating Tag shall backscatter 00000_2 followed by a PacketCRC.	M	Tag	**By design**
322	6.3.2.12.1.1	An Interrogator shall prepend a *Select* command with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
323	6.3.2.12.1.1	A Tag shall not reply to a *Select*.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
324	6.3.2.12.1.2	Interrogators and Tags may implement the *Challenge* command; if they do then they shall implement it as shown in Table 6.31.	O	Tag and Interrogator	**By design**
325	6.3.2.12.1.2	Upon receiving a *Challenge* a Tag that supports the command shall return to the **ready** state and deassert its **C** flag.	M	Tag	**By design**
326	6.3.2.12.1.2	If the Tag supports the CSI and can execute message then it shall perform the requested action(s); otherwise the Tag shall not execute message.	M	Tag	**By design**
327	6.3.2.12.1.2	A Tag shall not reply to a *Challenge*.	M	Tag	**By design**
328	6.3.2.12.1.2	A *Challenge* contains 2 RFU bits. An Interrogator shall set these bits to 002.	M	Interrogator	**By design**
329	6.3.2.12.1.2	If a Tag receives a *Challenge* containing nonzero RFU bits then it shall return to the **ready** state and deassert its **C** flag but not execute message.	M	Tag	**By design**
330	6.3.2.12.1.2	An Interrogator shall prepend a *Challenge* command with a frame-sync (see 6.3.1.2.8).	M	Interrogator	**By design**
331	6.3.2.12.1.2	If a Tag supports the *Challenge* command then it shall implement the security (**S**) indicator (see 6.3.2.1.3).	M	Tag	**By design**
332	6.3.2.12.1.2	After executing a *Challenge* a Tag shall store its response (result or error code) in its ResponseBuffer.	M	Tag	**By design**
333	6.3.2.12.1.2	After executing and storing a response a Tag shall assert its **C** flag.	M	Tag	**By design**
334	6.3.2.12.1.2	A Tag shall not assert its **C** flag until after it has computed and stored the entire response.	M	Tag	**By design**
335	6.3.2.12.1.2	A Tag shall deassert its **C** flag upon (a) receiving a subsequent *Challenge*, or (b) exceeding the **C** flag persistence time in Table 6.20.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
336	6.3.2.12.1.2	If the most recent *Challenge* received and executable by a Tag asserts immed, and if the Tag's **C** flag is asserted when it receives a subsequent *ACK*, then when replying to the *ACK* the Tag shall concatenate its ResponseBuffer contents to its EPC and backscatter the concatenated reply. See Table 6.17. See also Figure 6.23.	M	Tag	**By design**
337	6.3.2.12.1.2	If a Tag observes a properly formatted *Challenge* but there is a cryptographic error, and the cryptographic suite specifies that the error requires a security timeout, then the Tag shall return to **ready** and enforce a security timeout as specified in 6.3.2.5.	M	Tag	**By design**
338	6.3.2.12.1.2	If a Tag that supports security timeouts for a *Challenge* receives a *Challenge* during a timeout then it shall return to **ready** but not act on or otherwise execute any portion of the *Challenge*.	M	Tag	**By design**
339	6.3.2.12.2.1	Interrogators and Tags shall implement the *Query* command shown in Table 6.32.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
340	6.3.2.12.2.1	An Interrogator shall prepend a *Query* with a preamble (see 6.3.1.2.8).	M	Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
341	6.3.2.12.2.1	An Interrogator shall not encapsulate a *Query* in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interrogator	**By design**
342	6.3.2.12.2.1	If a Tag receives a *Query* with a CRC-5 error then it shall treat the command as invalid (see Table C.30).	M	Tag	**By design**
343	6.3.2.12.2.1	Upon receiving a *Query*, Tags with matching Sel and Target shall pick a random value in the range (0, 2^Q-1), inclusive, and shall load this value into their slot counter.	M	Tag	**By design**
344	6.3.2.12.2.1	If a Tag, in response to the *Query*, loads its slot counter with zero, then its reply to a *Query* shall be as shown in Table 6.33 using the *immediate* reply type specified in 6.3.1.6.1; otherwise the Tag shall remain silent.	M	Tag	**By design**
345	6.3.2.12.2.1	If a Tag in the **acknowledged**, **open**, or **secured** states receives a *Query* whose session parameter matches the prior session it shall invert its **inventoried** flag (i.e. *A*→*B* or *B*→*A*) for the session before it evaluates whether to transition to **ready**, **arbitrate**, or **reply**.	M	Tag	**By design**
346	6.3.2.12.2.1	If a Tag in the **acknowledged**, **open**, or **secured** states receives a *Query* whose session parameter does not match the prior session it shall leave its **inventoried** flag for the prior session unchanged when beginning the new round.	M	Tag	**By design**
347	6.3.2.12.2.1	Tags shall support all DR and M values specified in Table 6.9 and Table 6.10, respectively.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
348	6.3.2.12.2.1	A Tag in any state other than **killed** shall execute a *Query* command, starting a new round in the specified session and transitioning to **ready**, **arbitrate**, or **reply**, as appropriate (see Figure 6.21).	M	Tag	**By design**
349	6.3.2.12.2.1	A Tag in the **killed** state shall ignore a *Query*.	M	Tag	**By design**
350	6.3.2.12.2.2	Interrogators and Tags shall implement the *QueryAdjust* command shown in Table 6.34.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
351	6.3.2.12.2.2	If a Tag receives a *QueryAdjust* whose session number is different from the session number in the *Query* that initiated the round it shall ignore the command.	M	Tag	**By design**
352	6.3.2.12.2.2	If a Tag receives a *QueryAdjust* with an UpDn value different from those specified above it shall treat the command as invalid (see Table C.30).	M	Tag	**By design**
353	6.3.2.12.2.2	If a Tag whose *Q* value is 15 receives a *QueryAdjust* with UpDn = 110 then it shall change UpDn to 000 prior to executing the command; likewise, if a Tag whose *Q* value is 0 receives a *QueryAdjust* with UpDn = 011 then it shall change UpDn to 000 prior to executing the command.	M	Tag	**By design**
354	6.3.2.12.2.2	A Tag shall maintain a running count of the current *Q* value.	M	Tag	**By design**
355	6.3.2.12.2.2	An Interrogator shall prepend a *QueryAdjust* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
356	6.3.2.12.2.2	An Interrogator shall not encapsulate a *QueryAdjust* in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interrogat or	
357	6.3.2.12.2.2	If a Tag, in response to the *QueryAdjust*, loads its slot counter with zero, then its reply to a *QueryAdjust* shall be shown in Table 6.35 using the *immediate* reply type specified in 6.3.1.6.1; otherwise, the Tag shall remain silent.	M	Tag	**By design**
358	6.3.2.12.2.2	Tags shall respond to a *QueryAdjust* only if they received a prior *Query*.	M	Tag	**By design**
359	6.3.2.12.2.2	A Tag in any state except **ready** or **killed** shall execute a *QueryAdjust* command if, and only if, (i) the session parameter in the command matches the session parameter in the *Query* that started the round, and (ii) the Tag is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.12.3.4 or 6.3.2.12.3.6, respectively).	M	Tag	**By design**
360	6.3.2.12.2.2	A Tag in the **acknowledged**, **open**, or **secured** states that receive a *QueryAdjust* whose session parameter matches the session parameter in the prior *Query*, and that is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.12.3.4 or 6.3.2.12.3.6, respectively), shall invert its **inventoried** flag (i.e. *A*→*B* or *B*→*A*, as appropriate) for the current session and transition to **ready**.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
361	6.3.2.12.2.3	Interrogators and Tags shall implement the *QueryRep* command shown in Table 6.36.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
362	6.3.2.12.2.3	If a Tag receives a *QueryRep* whose session number is different from the session number in the *Query* that initiated the round it shall ignore the command.	M	Tag	**By design**
363	6.3.2.12.2.3	An Interrogator shall prepend a *QueryRep* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
364	6.3.2.12.2.3	An Interrogator shall not encapsulate a *QueryRep* in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interro-gator	
365	6.3.2.12.2.3	If a Tag, in response to the *QueryRep*, decrements its slot counter and the decremented slot value is zero, then its reply to a *QueryRep* shall be as shown in Table 6.37 using the *immediate* reply type specified in 6.3.1.6.1; otherwise the Tag shall remain silent.	M	Tag	**By design**
366	6.3.2.12.2.3	A Tag shall respond to a *QueryRep* only if they received a prior *Query*.	M	Tag	**By design**
367	6.3.2.12.2.3	A Tag in any state except **ready** or **killed** shall execute a *QueryRep* command if, and only if, (i) the session parameter in the command matches the session parameter in the *Query* that started the round, and (ii) the Tag is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.12.3.4 or 6.3.2.12.3.6, respectively).	M	Tag	**By design**
368	6.3.2.12.2.3	A Tag in the **acknowledged**, **open**, or **secured** state that receive a *QueryRep* whose session parameter matches the session parameter in the prior *Query*, and that is not in the middle of a *Kill* or *Access* command sequence (see 6.3.2.12.3.4 or 6.3.2.12.3.6, respectively), shall invert its **inventoried** flag (i.e. $A \rightarrow B$ or $B \rightarrow A$, as appropriate) for the current session and transition to **ready**.	M	Tag	**By design**
369	6.3.2.12.2.4	Interrogators and Tags shall implement the *ACK* command shown in Table 6.38.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
370	6.3.2.12.2.4	If an Interrogator issues an *ACK* to a Tag in the **reply** or **acknowledged** state, then the echoed RN16 shall be the RN16 that the Tag previously backscattered as it transitioned from the **arbitrate** state to the **reply** state.	M	Tag	**By design**
371	6.3.2.12.2.4	If an Interrogator issues an *ACK* to a Tag in the **open** or **secured** state, then the echoed RN16 shall be the Tag's handle (see 6.3.2.12.3.1).	M	Tag	**By design**
372	6.3.2.12.2.4	An Interrogator shall prepend an *ACK* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
373	6.3.2.12.2.4	The Tag reply to a successful *ACK* shall be as shown in Table 6.39, using the *immediate* reply type specified in 6.3.1.6.1.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
374	6.3.2.12.2.4	A Tag that receives an *ACK* with an incorrect RN16 or an incorrect handle (as appropriate) shall return to **arbitrate** without responding, unless the Tag is in **ready** or **killed**, in which case it shall ignore the *ACK* and remain in its current state.	M	Tag	**By design**
375	6.3.2.12.2.4	In either case a Tag's reply to an *ACK* shall not exceed 528 bits for the PC + EPC + PacketCRC, optionally followed by a response field and its associated CRC-16 (see Table 6.17).	M	Tag	**By design**
376	6.3.2.12.2.5	Interrogators and Tags shall implement the *NAK* command shown in Table 6.40.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
377	6.3.2.12.2.5	A Tag that receives a *NAK* shall return to the **arbitrate** state without changing its **inventoried** flag, unless the Tag is in **ready** or **killed**, in which case it shall ignore the *NAK* and remain in its current state.	M	Tag	**By design**
378	12	An Interrogator shall prepend a *NAK* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
379	6.3.2.12.2.5	An Interrogator shall not encapsulate a *NAK* in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interrogator	
380	6.3.2.12.2.5	A Tag shall not reply to a *NAK*.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
381	6.3.2.12.3	A Tag in the **open** or **secured** state that receives an access command with an incorrect handle but a correct CRC-16 shall behave as specified in Table C.30.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
382	6.3.2.12.3.1	Interrogators and Tags shall implement the *Req_RN* command shown in Table 6.41.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
383	6.3.2.12.3.1	When issuing a *Req_RN* to a Tag in the **acknowledged** state, an Interrogator shall include the Tag's last backscattered RN16 as a parameter in the *Req_RN*.	M	Interro-gator	**By design**
384	6.3.2.12.3.1	If a Tag receives a *Req_RN* with a correct RN16 and a correct CRC-16 then it shall generate and store a new RN16 (denoted handle), backscatter this handle, and transition to the **open** or **secured** state.	M	Tag	**By design**
385	6.3.2.12.3.1	A Tag in the **acknowledged** state that receives the *Req_RN* with an incorrect RN16 but a correct CRC-16 shall ignore the *Req_RN* and remain in the **acknowledged** state.	M	Tag	**By design**
386	6.3.2.12.3.1	When issuing a *Req_RN* to a Tag in the **open** or **secured** state, an Interrogator shall include the Tag's handle as a parameter in the *Req_RN*.	M	Interro-gator	**By design**
387	6.3.2.12.3.1	If a Tag receives the *Req_RN* with a correct handle and a correct CRC-16 then it shall generate and backscatter a new RN16, remaining in its current state (**open** or **secured**, as appropriate). .	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
388	6.3.2.12.3.1	The Tag that receives an *ACK* with a correct handle replies as specified in Table 6.39, whereas those that receive it with an incorrect handle shall return to **arbitrate** (Note: If a Tag receives an *ACK* with an incorrect handle it returns to **arbitrate**, whereas if it receives an access command with incorrect handle it behaves as specified n Table C.30.	M	Tag	**By design**
389	6.3.2.12.3.1	The first bit of the backscattered RN16 shall be denoted the MSB; the last bit shall be denoted the LSB.	M	Tag	**By design**
390	6.3.2.12.3.1	An Interrogator shall prepend a *Req_RN* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
391	6.3.2.12.3.1	A Tag's reply to a *Req_RN* shall be as shown in Table 6.42 using the *immediate* reply type specified in 6.3.1.6.1.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
392	6.3.2.10.3.2	Interrogators and Tags shall implement the *Read* command shown in Table 6.44.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
393	6.3.2.10.3.2	*Read* commands shall apply to a single memory bank.	M	Tag	**By design**
394	6.3.2.10.3.2	If WordCount = 00_h then a Tag shall backscatter the contents of the chosen memory bank starting at WordPtr and ending at the end of the memory bank or file, however: • if MemBank = 01_2, then a Tag shall backscatter the memory contents specified in Table 6.43. • if MemBank=10_2, and part of TID memory is untraceably hidden (see 6.3.2.12.3.16), and the Interrogator has a deasserted Untraceable privilege, and the memory address specified by WordPtr is in the traceable part of TID memory, then a Tag may either (i) backscatter the traceable part of TID memory starting at WordPtr, or (ii) treat the command's parameters as unsupported (see Table C.30), depending on the Tag manufacturer's implementation.	M	Tag	**By design**
395	6.3.2.10.3.2	An Interrogator shall prepend a Read with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
396	6.3.2.10.3.2	An unauthenticated Interrogator may, and an authenticated Interrogator shall, encapsulate a *Read* command in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interrogat or	**By design**
397	6.3.2.10.3.2	A Tag shall reply to a *Read* using the *immediate* reply type specified in 6.3.1.6.1.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
398	6.3.2.10.3.2	If all of the memory words specified in a *Read* exist and none are read-locked, all are traceable or the Interrogator has an asserted Untraceable privilege, and for User memory the Interrogator has read privileges to the currently open file (see 6.3.2.11.3), then a Tag's reply to a *Read* shall be as shown in Table 6.45 comprising a header (a 0-bit), the requested memory words, and the Tag's handle.	M	Tag	**By design**
399	6.3.2.10.3.2	Otherwise the Tag shall not execute the *Read* and instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By design**
400	6.3.2.12.3.3	Interrogators and Tags shall implement the *Write* command shown in Table 6.46.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
401	6.3.2.12.3.3	*Write* commands shall apply to a single memory bank.	M	Tag and Interrogator	**By design**
402	6.3.2.12.3.3	Before each and every *Write* the Interrogator shall first issue a *Req_RN* command; the Tag replies by backscattering a new RN16.	M	Interro-gator	**By design**
403	6.3.2.12.3.3	The Interrogator shall cover code the data by EXORing it with this new RN16 prior to transmission.	M	Interro-gator	**By design**
404	6.3.2.12.3.3	A Tag shall only execute a *Write* in the **open** or **secured** state.	M	Tag	**By design**
405	6.3.2.12.3.3	If a Tag in the **open** or **secured** states receives a *Write* before which the immediately preceding command was not a *Req_RN* then it shall not execute the *Write* and instead treat the command as invalid (see Table C.30).	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
406	6.3.2.12.3.3	If an Interrogator attempts to write to the kill or access password, EPC or TID memory banks, or File_0 and these memory locations are permalocked; or to the kill or access password, EPC or TID memory banks, or File_0 and these memory locations are locked unwriteable and the Tag is in the **open** state; or to a permalocked block in File_N, N>0 of User memory; or to memory that is untraceably hidden and the Interrogator has a deasserted Untraceable privilege; or to a file for which the Interrogator does not have sufficient privileges; then the Tag shall not execute the *Write* and instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By demonstration – see state machine** 1. Attempt to write the PC value in open state, while the EPC memory is locked. Verify that the Tag replies with an error message and that the memory has not been written. 2. Attempt to write a permalocked memory location from secured state. Verify that the Tag replies with an error message and that the memory has not been written. 3. Attempt to write a permalocked block in File_N, N>0 of USER memory. Verify that the Tag replies with an error message and that the memory has not been written. 4. Attempt to write to a memory location that is untraceably hidden with a deasserted Untracable privilege. Verify that the Tag replies with an error message and that the memory has not been written. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
407	6.3.2.12.3.3	An Interrogator shall prepend a *Write* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
408	6.3.2.12.3.3	An Interrogator shall not encapsulate a *Write* in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interrogator	**By design**
409	6.3.2.12.3.3	Upon receiving a valid *Write* command a Tag shall write the commanded Data into memory.	M	Tag	**By design**
410	6.3.2.12.3.3	A Tag shall reply to a *Write* using the *delayed* reply specified in 6.3.1.6.2.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
411	6.3.2.12.3.4	Interrogators and Tags shall implement the *Kill* command shown in Table 6.47.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
412	6.3.2.12.3.4	To kill a Tag, an Interrogator shall follow the kill procedure shown in Figure 6.24.	M	Interro-gator	**By design**
413	6.3.2.12.3.4	A Tag shall implement the password-based kill sequence shown in the left-side branch of the kill procedure in Figure 6.24.	M	Tag	**By design**
414	6.3.2.12.3.4	An Interrogator shall set these bits to 000_2.	M	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
415	6.3.2.12.3.4	A Tag shall ignore these bits.	M	Tag	**By design**
416	6.3.2.12.3.4	An Interrogator shall prepend an unencapsulated *Kill* command with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
417	6.3.2.12.3.4	Each EXOR operation shall be performed MSB first (i.e. the MSB of each half-password shall be EXORed with the MSB of its respective RN16).	M	Interro-gator	**By design**
418	6.3.2.12.3.4	A Tag shall be capable of successively accepting two 16-bit subportions of the 32-bit kill password.	M	Tag	**By design**
419	6.3.2.12.3.4	An Interrogator shall not intersperse commands other than *Req_RN* between the two successive *Kill* commands.	M	Interro-gator	**By design**
420	6.3.2.12.3.4	If a Tag, after receiving a first *Kill*, receives any valid command other than *Req_RN* before the second *Kill* then it shall not execute the command and instead treat is as improper (see Table C.30), unless the intervening command is a *Query*, in which case the Tag shall execute the *Query* and invert its **inventoried** flag if the session parameter in the *Query* matches that in the prior session.	M	Tag	**By design**
421	6.3.2.12.3.4	A Tag with a zero-valued kill password shall disallow itself from being killed by a password-based kill operation.	M	Tag	**By demonstration** Tested in compliance with Item 139.
422	6.3.2.12.3.4	A Tag with a zero-valued kill password shall respond to a password-based kill by not executing the kill operation and backscattering an error code, remaining in its current state. See Figure 6.24.	M	Tag	**By demonstration** Tested in compliance with Item 139.
423	6.3.2.12.3.4	A Tag shall reply to a first *Kill* using the *immediate* reply specified in 6.3.1.6.1.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
424	6.3.2.12.3.4	The Tag`s first reply shall be as shown in Table 6.48.	M	Tag	**By design**
425	6.3.2.12.3.4	The reply shall use the TRext value specified in the *Query* command that initiated the round.	M	Tag	**By design**
426	6.3.2.12.3.4	A Tag shall reply to the second *Kill* using the *delayed* reply specified in 6.3.1.6.2.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
427	6.3.2.12.3.4	If the kill succeeds then the Tag, after sending the final reply shown in Table 6.13, shall render itself silent and shall not respond to an Interrogator thereafter.	M	Tag	**By design**
428	6.3.2.12.3.4	If a Tag observes a properly formatted password-based *Kill*-command sequence but the kill fails (as will happen if the Interrogator sends an incorrect kill password) then the Tag shall return to **arbitrate** and may enforce a security timeout as specified in 6.3.2.5.	M, O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
429	6.3.2.12.3.4	If a Tag that supports security timeouts for a password-based *Kill*-command sequence receives such a sequence during a timeout then it shall behave as though it is not killable, backscatter an error code (see Annex I), and remain in its current state.	O	Tag	**By design**
430	6.3.2.12.3.4	A Tag shall authenticate an Interrogator via na Interrogator or mutual authentication prior to executing an authenticated kill.	O	Tag	**By design**
431	6.3.2.12.3.4	The Interrogator may use any 16-bit value in the password field of the *Kill* command because a Tag shall ignore the kill password for an authenticated kill.	O	Tag	**By design**
432	6.3.2.12.3.4	A Tag shall only execute an authenticated kill if the Interrogator possesses an asserted AuthKill privilege (see Table 6.23) and the Tag is in the **secured** state.	O	Tag	**By design**
433	6.3.2.12.3.4	A Tag shall reply to an authenticated kill using an in-process reply (as required by a *SecureComm* or *AuthComm*), but with SenRep=1 regardless of the SenRep value actually specified in the *SecureComm* or *AuthComm*.	O	Tag	**By design**
434	6.3.2.12.3.4	If the kill succeeds then the Tag, after sending the final reply shown in Table 6.13, shall transition to the **killed** state and not respond to an Interrogator thereafter.	O	Tag	**By design**
435	6.3.2.12.3.4	If the kill fails then the Tag shall remain in its current state and backscatter an error code (see Annex I), unless the Tag is in the **open** state, the Interrogator is not authenticated, or the Interrogator does not have an asserted AuthKill privilege (see Table 6.23), in which case the Tag shall return to **arbitrate** and may enforce a security timeout as specified in 6.3.2.5.	O	Tag	**By design**
436	6.3.2.12.3.4	If a Tag that supports security timeouts for an authenticated *Kill* command receives an authenticated *Kill* command during a timeout then it shall behave as though it is not killable, backscatter an error code (see Annex I), and remain in its current state.	O	Tag	**By design**
437	6.3.2.12.3.5	Interrogators and Tags shall implement the *Lock* command shown in Table 6.49 and Figure 6.25.	M	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
438	6.3.2.12.3.5	A Tag shall interpret these bit values as follows: Mask = 0: Ignore the associated Action field and retain the current lock setting; Mask = 1: Implement the associated Action field and overwrite the current lock setting.	M	Tag	**By design**
439	6.3.2.12.3.5	A Tag shall interpret these bit values as follows: Action = 0: Deassert lock for the associated memory location; Action = 1: Assert lock or permalock for the associated memory location.	M	Tag	**By design**
440	6.3.2.12.3.5	The payload of a *Lock* command shall always be 20 bits in length.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
441	6.3.2.12.3.5	If an Interrogator issues a *Lock* command whose Mask and Action fields attempt to change the lock status of a nonexistent memory bank, nonexistent File_0, or nonexistent password, then the Tag shall shall not execute the *Lock* command and instead treat the command`s parameters as unsupported (see Table C.30).	M	Tag	**By design**
442	6.3.2.12.3.5	If a Tag receives a *Lock* whose payload attempts to deassert a previously asserted permalock bit, then the Tag shall shall not execute the *Lock* command and instead treat the command`s parameters as unsupported (see Table C.30).	M	Tag	**By design**
443	6.3.2.12.3.5	If a Tag receives a *Lock* whose payload attempts to reassert a previously asserted permalock bit, then the Tag shall simply ignore this particular Action field and implement the remainder of the *Lock* payload.	M	Tag	**By design**
444	6.3.2.12.3.5	Regardless of the location, a field-deployed Tag shall not permit an Interrogator to change its lock bits except by means of a *Lock* command	M	Tag	**By design**
445	6.3.2.12.3.5	A Tag shall implement memory locking and the *Lock* command.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
446	6.3.2.10.3.5	If a Tag receives a *Lock* it cannot execute because one or more memory locations do not exist, or one or more of the Action fields attempt to change a previously permalocked value, or one or more of the memory locations are either not lockable or not unlockable, the Tag shall not execute the *Lock* and instead instead treat the command's parameters as unsupported (see Table C.30).	M	Tag	**By design**
447	6.3.2.10.3.5	The only exception to this general rule is for a Tag that (a) does not support File_N, N>0 and (b) whose only lock functionality is to permanently lock **all** memory (i.e. all memory banks and all passwords) at once; such a Tag shall execute a *Lock* whose payload is FFFFFh, and shall backscatter an error code for any payload other than FFFFFh.	M	Tag	**By design**
448	6.3.2.12.3.5	A Tag in the **secured** state shall permit an Interrogator to write or erase memory locations with (pwd-write=1 AND permalock=0) or (pwd-read/write=1 AND permalock=0) without first issuing a *Lock* to change these fields.	M	Tag	**By design**
449	6.3.2.12.3.5	An Interrogator shall prepend a *Lock* with a frame-sync (see 6.3.1.2.8).	M	Interro-gator	**By design**
450	6.3.2.12.3.5	An unauthenticated Interrogator may, and an authenticated Interrogator shall, encapsulate a *Lock* command in a *SecureComm* or *AuthComm* (see Table 6.28).	M	Interrogator	**By design**
451	6.3.2.12.3.5	Upon receiving a valid *Lock* command a Tag shall perform the commanded lock operation.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
452	6.3.2.12.3.5	A Tag shall reply to a *Lock* using the *delayed* reply specified in 6.3.1.6.2.	M	Tag	**By design** Also tested in compliance with 6.3.2.6, Figure 6.21
453	6.3.2.12.3.6	Interrogators and Tags may implement an *Access* command; if they do, they shall implement it as shown in Table 6.51.	O	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
454	6.3.2.12.3.6	To access a Tag, an Interrogator shall follow the multi-step procedure outlined in Figure 6.25.	O	Interro-gator	**By design**
455	6.3.2.12.3.6	Each EXOR operation shall be performed MSB first (i.e. the MSB of each half-password shall be EXORed with the MSB of its respective RN16).	O	Interro-gator	**By design**
456	6.3.2.12.3.6	A Tag shall be capable of successively accepting two 16-bit subportions of the 32-bit access password.	O	Tag	**By design**
457	6.3.2.12.3.6	An Interrogator shall not intersperse commands other than a *Req_RN* between the two successive *Access* commands.	O	Interro-gator	**By design**
458	6.3.2.12.3.6	If a Tag, after receiving a first *Access*, receives any valid command other than *Req_RN* before the second *Access* then it shall not execute the command and instead treat it as improper (see Table C.30), unless the intervening command is a *Query*, in which case the Tag shall execute the *Query* and invert its **inventoried** flag if the session parameter in the *Query* matches that in the prior session.	O	Tag	**By design**
459	6.3.2.12.3.6	An Interrogator shall prepend an *Access* with a frame-sync (see 6.3.1.2.8).	O	Interro-gator	**By design**
460	6.3.2.12.3.6	An Interrogator shall not encapsulate an *Access* in a *SecureComm* or *AuthComm* (see Table 6.28).	O	Interrogat or	**By design**
461	6.3.2.12.3.6	A Tag shall reply to an *Access* using the *immediate* reply specified in 6.3.1.6.1.	O	Tag	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
462	6.3.2.10.3.7	Interrogators and Tags may implement a *BlockWrite* command; if they do, they shall implement it as shown in Table 6.42.	O, TACF	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
463	6.3.2.10.3.7	*BlockWrite* commands shall apply to a single memory bank.	O, TACF	Tag	**By design**
464	6.3.2.10.3.7	If WordCount = 00_h then a Tag shall treat the *BlockWrite as invalid*.	O, TACF	Tag	**By design**
465	6.3.2.10.3.7	If WordCount = 01_h then a Tag shall write a single data word.	O, TACF	Tag	**By design**
466	6.3.2.10.3.7	Data contains the 16-bit words to be written, and shall be 16 × WordCount bits in length.	O, TACF	Interro-gator	**By design**
467	6.3.2.10.3.7	A Tag shall only execute a *BlockWrite* in the **open** or **secured** state.	O, TACF	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
468	6.3.2.10.3.7	If an Interrogator attempts to write to the kill or access password, EPC or TID memory banks, or File_0 and these memory locations are permalocked; or to the kill or access password, EPC or TID memory banks, or File_0 and these memory locations are locked unwriteable and the Tag is in the **open** state; or to memory that is untraceably hidden and the Interrogator has a deasserted Untraceable privilege; or to a file for which the Interrogator does not have sufficient privileges; or if WordPtr and WordCount include one or more permalocked blocks in File_N, N≥0 of User memory; then the Tag shall not execute the *BlockWrite* and instead treat the command's parameters as unsupported(see Table C.30).	O, TACF	Tag	**By design**
469	6.3.2.10.3.7	An interrogator shall prepend a *BlockWrite* with a frame-sync (see 6.3.1.2.8).	O, TACF	Interro-gator	**By design**
470		An unauthenticated Interrogator may, and an authenticated Interrogator shall, encapsulate a *BlockWrite* in a *SecureComm* or *AuthComm* (see Table 6.28).	O, TACF	Interrogator	
471	6.3.2.10.3.7	Upon receiving a valid *BlockWrite* command a Tag shall write the commanded data into memory.	O, TACF	Tag	**By design**
472	6.3.2.10.3.7	A Tag reply to a *BlockWrite* shall using the delayed reply specified in 6.3.1.6.2.	O, TACF	Tag	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
473	6.3.2.10.3.8	Interrogators and Tags may implement a *BlockErase* command; if they do, they shall implement it as shown in Table 6.44.	O	Tag and Interro-gator	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
474	6.3.2.10.3.8	*BlockErase* commands shall apply to a single memory bank.	O	Tag	**By design**
475	6.3.2.10.3.8	If WordCount = 00_h the Tag shall treat the *BlockErase as invalid*.	O	Tag	**By design**
476	6.3.2.10.3.8	If WordCount = 01_h then a Tag shall erase a single data word.	O	Tag	**By design**
477	6.3.2.10.3.8	A Tag shall only execute a *BlockErase* in the **open** or **secured** state.	O	Tag	**By design**
478		If an Interrogator attempts to erase the kill or access password, EPC or TID memory banks, or File_0 and these memory locations are permalocked; or the kill or access password, EPC or TID memory banks, or File_0 and these memory locations are locked unwriteable and the Tag is in the **open** state; or to memory that is untraceably hidden and the Interrogator has a deasserted Untraceable privilege; or a file for which the Interrogator does not have sufficient privileges; or if WordPtr and WordCount include one or more permalocked blocks in File_N, N≥0 of User memory; then the Tag shall not execute the *BlockErase* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
479	6.3.2.10.3.8	An Interrogator shall prepend a *BlockErase* with a frame-sync (see 6.3.1.2.8).	O	Interro-gator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
480	6.3.2.10.3.8	Upon receiving an executable *BlockErase* command a Tag shall erase the commanded memory words.	O	Tag	**By design**
481	6.3.2.10.3.8	A Tag shall reply to a *BlockErase* using the *delayed reply* specified in 6.3.1.6.2.	O	Tag	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
482		An unauthenticated Interrogator may, and an authenticated Interrogator shall, encapsulate a *BlockErase* in a *SecureComm* or *AuthComm* (see Table 6.28).	O	Interrogator	**By design**
483	6.3.2.10.3.9	Interrogators and Tags may implement a *BlockPermalock* command; if they do, they shall implement it as shown in Table 6.56. *BlockPermalock* allows an Interrogator to:	O, CE	Tag and Interro-gator	**By demonstration** Issue a *BlockPermalock* command to permalock a single block of any File_N, N>0. After successful tag response, attempt to write the selected block to verify that it has been locked. The Tag shall not execute the *Write* and instead treat the command's parameters as unsupported Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
484	6.3.2.10.3.9	A Tag shall only execute a *BlockPermalock* in the **secured** state.	O, CE	Tag	**By design**
485	6.3.2.10.3.9	Table 6.55 specifies how a Tag shall behave upon receiving a *BlockPermalock* targeting File_0 that follows a prior *Lock*, or vice versa (assuming Read/Lock=1).	O, CE	Tag	**By design**
486	6.3.2.10.3.9	MemBank specifies whether the *BlockPermalock* applies to EPC, TID, or User memory. *BlockPermalock* commands shall apply to a single memory bank. Successive *BlockPermalocks* may apply to different memory banks.	O, CE	Tag and Interrogator	**By design**
487	6.3.2.10.3.9	A Tag shall only execute a *BlockPermalock* command if MemBank=11 (User memory); if a Tag receives a *BlockPermalock* with MemBank<>11 then it shall not execute the *BlockPermalock* and instead treat the command's parameters as unsupported (see Table C.30). Future protocols may use these other MemBank values to expand the functionality of the *BlockPermalock* command.	O, CE	Tag	**By design**
488	6.3.2.10.3.9	A Tag shall interpret the Read/Lock bit as follows:	O, CE	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
489	6.3.2.10.3.9	Read/Lock=0: A Tag shall backscatter the permalock status of blocks in the specified memory bank, starting from the memory block located at BlockPtr and ending at the memory block located at BlockPtr+(16×BlockRange)–1.	O, CE	Tag	**By design**
490	6.3.2.10.3.9	A Tag shall backscatter a "0" if the memory block corresponding to that bit is not permalocked and a "1" if the block is permalocked. An Interrogator omits Mask from the *BlockPermalock* when Read/Lock=0.	O, CE	Tag	**By design**
491	6.3.2.10.3.9	Read/Lock=1: A Tag shall permalock those blocks in the specified memory bank that are specified by Mask, starting at BlockPtr and ending at BlockPtr+(16×BlockRange)–1.	O, CE	Tag	**By design**
492	6.3.2.10.3.9	If BlockRange=00_h then a Tag shall not execute the *BlockPermalock* and instead treat the command's parameters as unsupported (see Table C.30).	O, CE	Tag	**By design**
493	6.3.2.10.3.9	Read/Lock=0: The Interrogator shall omit Mask from the *BlockPermalock*.	O, CE	Interrogator	**By design**
494	6.3.2.10.3.9	Read/Lock=1: The Interrogator shall include a Mask of length 16×BlockRange bits in the *BlockPermalock*.	O, CE	Interrogator	**By design**
495	6.3.2.10.3.9	The Mask bits shall be ordered from lower-order block to higher (i.e. if BlockPtr=00_h then the leading Mask bit refers to block 0).	O, CE	Tag and Interrogator	**By design**
496	6.3.2.10.3.9	The Tag shall interpret each bit of Mask as follows:	O, CE	Tag	**By demonstration** Tested in compliance with Item 484.
497	6.3.2.10.3.9	Mask bit=1: Permalock the corresponding memory block. If a block is already permalocked then a Tag shall retain the current permalock setting. A memory block, once permalocked, cannot be un-permalocked.	O, CE	Tag and Interrogator	**By demonstration** Tested in compliance with Item 484.
498	6.3.2.10.3.9	A *BlockPermalock* contains 8 RFU bits. An Interrogator shall set these bits to 00_h.	O, CE	Interrogator	**By demonstration** Tested in compliance with Item 484.
499	6.3.2.10.3.9	A Tag in the **secured** state that receives a *BlockPermalock* with nonzero RFU bits shall not execute the *BlockPermalock* and instead treat the command's parameters as unsupported (see Table C.30). Future protocols may use these RFU bits to expand the *BlockPermalock* command's functionality.	O, CE	Tag	**By design**
500	6.3.2.10.3.9	If a Tag receives a *BlockPermalock* that it cannot execute because User memory does not exist, or User memory is untraceably hidden and the Interrogator has a deasserted Untraceable privilege, or in which one of the asserted Mask bits references a non-existent memory block, or because the Interrogator has insufficient file privileges (see 6.3.2.11.3) then the Tag shall not execute the *BlockPermalock* and instead treat the command's parameters as unsupported (see Table C.30).	O, CE	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
501	6.3.2.10.3.9	A Tag shall treat as invalid a *BlockPermalock* in which Read/Lock=0 but Mask is not omitted, or a *BlockPermalock* in which Read/Lock=1 but Mask has a length not equal to 16×BlockRange bits (see Table C.30).	O, CE	Tag	**By design**
502	6.3.2.10.3.9	Certain Tags, depending on the Tag manufacturer's implementation, may be unable to execute a *BlockPermalock* with certain BlockPtr and BlockRange values, in which case the Tag shall not execute the *BlockPermalock* and instead treat the command's parameters as unsupported (see Table C.30).	O, CE	Tag	**By design**
503	6.3.2.10.3.9	If an Interrogator issues a *BlockPermalock* in which BlockPtr and BlockRange specify one or more nonexistent blocks, but Mask only asserts permalocking on existent blocks, then the Tag shall execute the *BlockPermalock*.	O, CE	Tag	**By design**
504	6.3.2.10.3.9	An Interrogator shall prepend a *BlockPermalock* with a frame-sync (see 6.3.1.2.8).	O, CE	Interrogator	**By design**
505	6.3.2.10.3.9	An unauthenticated Interrogator may, and an authenticated Interrogator shall, encapsulate a *BlockPermalock* in a *SecureComm* or *AuthComm* (see Table 6.28).	O, CE	Interrogator	**By design**
506	6.3.2.10.3.9	Upon receiving an executable *BlockPermalock* a Tag shall perform the requested operation, unless the Tag does not support block permalocking in which case it shall treat the command as invalid (see Table C.30).	O, CE	Tag	**By design**
507	6.3.2.10.3.9	If Read/Lock=0 then a Tag shall reply to a *BlockPermalock* using the *immediate* reply type specified in 6.3.1.6.1.	O, CE	Tag	**By design**
508	6.3.2.10.3.9	If the Tag is able to execute the *BlockPermalock* then its reply shall be as shown in Table 6.57 comprising a header (a 0-bit), the requested permalock bits, and the Tag's handle.	O, CE	Tag	**By design**
509	6.3.2.10.3.9	If the Tag is unable to execute the *BlockPermalock* then it shall backscatter an error code (see Table C.30, unsupported parameters) rather than the reply shown in Table 6.57.	O, CE	Tag	**By design**
510	6.3.2.10.3.9	The Tag's reply when Read/Lock=0 shall use the preamble specified by the TRext value in the *Query* that initiated the inventory round.	O, CE	Tag	**By design**
511	6.3.2.10.3.9	If Read/Lock=1 then a Tag shall reply to a *BlockPermalock* using the *delayed* reply specified in 6.3.1.6.2.	O, CE	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
512	6.3.2.12.3.10	Interrogators and Tags may implement the *Authenticate* command; if they do, they shall implement it as shown in Table 6.58.	O, TACA, TACF	Tag and Interrogator	**By demonstration** Issue an Authenticate command configured according a supported cryptographic suite. Verify that the Tag backscatters a valid response. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
513	6.3.2.12.3.10	An *Authenticate* contains 2 RFU bits. An Interrogator shall set these bits to 00_2.	O, TACA, TACF	Interrogatorr	**By design**
514	6.3.2.12.3.10	A Tag in the **open** or **secured** states that receives an *Authenticate* with nonzero RFU bits shall not execute the *Authenticate* and instead treat the command's parameters as unsupported (see Table C.30).	O, TACA, TACF	Tag	**By design**
515	6.3.2.12.3.10	An Interrogator shall prepend an *Authenticate* with a frame-sync (see 6.3.1.2.8).	O, TACA, TACF	Interrogator	**By design**
516	6.3.2.12.3.10	An Interrogator shall not encapsulate an *Authenticate* in a *SecureComm* or *AuthComm* (see Table 6.28).	O, TACA, TACF	Interrogator	**By design**
517	6.3.2.12.3.10	If a Tag supports the *Authenticate* command then it shall implement the security (**S**) indicator (see 6.3.2.1.3).	O, TACA, TACF	Tag	**By design**
518	6.3.2.12.3.10	An *Authenticate* command shall use the *in-process* reply specified in 6.3.1.6.3.	O, TACA	Tag	**By design**
519	6.3.2.12.3.10	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 (see Table C.30)	O, TACA, TACF	Tag	**By design**
520	6.3.2.12.3.10	If a Tag in the **secured** state receives an *Authenticate* that begins a new authentication, such as if the *Authenticate* contains a changed CSI, then the Tag shall transition to the **open** state, discontinue using and reset the current cryptographic engine, and begin the new authentication.	O, TACA, TACF	Tag	**By design**
521	6.3.2.12.3.10	If a Tag receives a properly formatted *Authenticate* but there is a cryptographic error, and the cryptographic suite specifies that the error requires a security timeout, then the Tag shall set a security timeout as specified in 6.3.2.5.	O, TACA, TACF	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
522	6.3.2.12.3.10	If a Tag that supports security timeouts for the *Authenticate* command receives an *Authenticate* during a timeout then it shall reject the command, backscatter an error code (see Annex I), and remain in its current state.	O TACA, TACF	Tag	**By design**
523	6.3.2.12.3.11	Interrogators and Tags may implement the *AuthComm* command; if they do, they shall implement it as shown in Table 6.59.	O	Tag, Interrogator	**By demonstration** Issue an AuthComm command configured according a supported cryptographic suite. Verify that the Tag backscatters a valid response. Verify that the interrogator decodes the tag response according the specification. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
524	6.3.2.12.3.11	An *AuthComm* shall always be preceded by a Tag, Interrogator, or mutual authentication via an *Authenticate* or a *Challenge*.	O	Tag, Interrogator	**By design**
525	6.3.2.12.3.11	An Interrogator shall remove the command's preamble, handle, and CRC before encapsulating it in an *AuthComm*.	O	Interrogator	**By design**
526	6.3.2.12.3.11	The encapsulated command shall not be encrypted or obscured.	O	Tag Interrogator	**By design**
527	6.3.2.12.3.11	An Interrogator shall set these bits to 00_2	O	Tag Interrogator	**By design**
528	6.3.2.12.3.11	A Tag in the **open** or **secured** states that receives an *AuthComm* with nonzero RFU bits shall not execute the *AuthComm* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
529	6.3.2.12.3.11	A Tag in the **open** or **secured** states that receives an *AuthComm* encapsulating a disallowed command, an unsupported command, or a command that does not support encapsulation (see Table 6.28) shall not execute the *AuthComm* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
530	6.3.2.12.3.11	An Interrogator shall prepend an *AuthComm* with a frame-sync (see 6.3.1.2.8).	O	Interrogator	**By design**
531	6.3.2.12.3.11	A Tag shall only accept an *AuthComm* after a successful cryptographic authentication.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
532		Because an *Access* command sequence is not a cryptographic authentication, a Tag that most recently entered the **secured** state via a successful *Access* command sequence shall not execute an *AuthComm* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
533	6.3.2.12.3.11	When processing an *AuthComm* a Tag shall first perform the functions/analysis/state-change/error-handling for the *AuthComm* itself and then, if the *AuthComm* is successful, the functions/analysis/state-change/error-handling for the command encapsulated in the *AuthComm*'s message field.	O	Tag	**By design**
534	6.3.2.12.3.11	A Tag shall reply to an *AuthComm* using the *in-process* reply specified in 6.3.1.6.3.	O	Tag	**By design**
535	6.3.2.12.3.11	The cryptographic suite shall specify the parameters that a Tag includes in its response, including at least the reply for the encapsulated command minus preamble, handle, and CRC.	O	???	**By design**
536	6.3.2.12.3.11	Unlike other commands that use an *in-process* reply, *AuthComm* does not include a SenRep field because a Tag shall always send (i.e. never store) its reply to an *AuthComm*.	O	Tag	**By design**
537	6.3.2.12.3.11	If a Tag receives a properly formatted *AuthComm* but there is a cryptographic error, and the cryptographic suite specifies that the error requires a security timeout, then the Tag shall set a security timeout as specified in 6.3.2.5.	O	Tag	**By design**
538	6.3.2.12.3.11	If a Tag that supports security timeouts for the *AuthComm* command receives an *AuthComm* during a timeout then it shall reject the command, backscatter an error code (see Annex I), and remain in its current state.	O	Tag	**By design**
539	6.3.2.12.3.12	Interrogators and Tags may implement the *SecureComm* command; if they do, they shall implement it as shown in Table 6.60.	O, TACF	Tag and Interogator	**By demonstration** Issue a SecureComm command configured according a supported cryptographic suite. Verify that the Tag backscatters a valid response. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
540	6.3.2.12.3.12	A *SecureComm* shall always be preceded by a Tag, Interrogator, or mutual authentication via an *Authenticate* or a *Challenge*.	O, TACF	Interrogator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
541	6.3.2.12.3.12	An Interrogator shall remove the command's preamble, handle, and CRC before encapsulating it in a *SecureComm*.	O, TACF	Interrogator	**By design**
542	6.3.2.12.3.12	The encapsulated command shall be encrypted.	O, TACF	Interrogator	**By design**
543	6.3.2.12.3.12	An Interrogator shall set these bits to 00_2.	O, TACF	Interrogator	**By design**
544	6.3.2.12.3.12	A Tag in the **open** or **secured** states that receives a *SecureComm* with nonzero RFU bits shall not execute the *SecureComm* and instead treat the command's parameters as unsupported (see Table C.30).	O, TACF	Tag	**By design**
545	6.3.2.12.3.12	A Tag in the **open** or **secured** states that receives a *SecureComm* encapsulating a disallowed command, an unsupported command, or a command that does not support encapsulation (see Table 6.28) shall not execute the *SecureComm* and instead treat the command's parameters as unsupported (see Table C.30).	O, TACF	Tag	**By design**
546	6.3.2.12.3.12	An Interrogator shall prepend a *SecureComm* with a frame-sync (see 6.3.1.2.8).	O, TACF	Tag	**By design**
547	6.3.2.12.3.12	A Tag shall only accept a *SecureComm* after a successful cryptographic authentication.	O, TACF	Tag	**By design**
548	6.3.2.12.3.12	Because an *Access* command sequence is not a cryptographic authentication, a Tag that most recently entered the **secured** state via a successful *Access* command sequence shall not execute a *SecureComm* and instead treat the command's parameters as unsupported (see Table C.30).	O, TACF	Tag	**By design**
549	6.3.2.12.3.12	When processing a *SecureComm* a Tag shall first perform the functions/analysis/state-change/error-handling for the *SecureComm* itself and then, if the *SecureComm* is successful, the functions/analysis/state-change/error-handling for the command encapsulated in the *SecureComm*'s message field.	O, TACF	Tag	**By design**
550	6.3.2.12.3.12	A Tag shall reply to a *SecureComm* using the *in-process* reply specified in 6.3.1.6.3.	O, TACF	Tag	**By design**
551	6.3.2.12.3.12	The cryptographic suite shall specify the parameters that a Tag includes in its response, including at least the reply for the encapsulated command minus preamble, handle, and CRC.	O, TACF	Tag	**By design**
552	6.3.2.12.3.12	If a Tag receives a properly formatted *SecureComm* but there is a cryptographic error, and the cryptographic suite specifies that the error requires a security timeout, then the Tag shall set a security timeout as specified in 6.3.2.5.	O, TACF	Tag	**By design**
553	6.3.2.12.3.12	If a Tag that supports security timeouts for the *SecureComm* command receives a *SecureComm* during a timeout then it shall reject the command, backscatter an error code (see Annex I), and remain in its current state.	O, TACF	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
554	6.3.2.12.3.13	Interrogators and Tags may implement the *KeyUpdate* command; if they do, they shall implement it as shown in Table 6.61.	O, TACF	Tag and Interrogator	**By demonstration** Issue a KeyUpdate command configured according a supported cryptographic suite. Verify that the Tag backscatters a valid response. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
555	6.3.2.12.3.13	A *KeyUpdate* shall always be preceded by an Interrogator or mutual authentication via an *Authenticate*.	O, TACF	Interrogator	**By design**
556	6.3.2.12.3.13	An Interrogator shall set these bits to 00_2.	O, TACF	Interrogator	**By design**
557	6.3.2.12.3.13	A Tag in the **secured** state that receives a *KeyUpdate* with nonzero RFU bits shall not execute the *KeyUpdate* and instead treat the command's parameters as unsupported (see Table C.30).	O, TACF	Tag	**By design**
558	6.3.2.12.3.13	If a cryptographic suite allows sending a *KeyUpdate* in an *AuthComm* or without encapsulation then message in the *KeyUpdate* shall be encrypted.	O, TACF	Interrogator	**By design**
559	6.3.2.12.3.13	A Tag in the **secured** state shall only write a key if (a) the Interrogator authenticated itself as a crypto superuser and KeyID is assigned to the same cryptographic suite as that specified by CSI in the *Authenticate* command that preceded the *KeyUpdate*, or (b) KeyID is the same as that used by the Interrogator to authenticate itself.	O, TACF	Tag	**By design**
560	6.3.2.12.3.13	In all other instances the Tag shall not execute the *KeyUpdate* and instead treat the command's parameters as unsupported (see Table C.30). See 6.3.2.11.2 for a description of Tag privileges and the crypto superuser privilege.	O, TACF	Tag	**By design**
561	6.3.2.12.3.13	Upon receiving an executable *KeyUpdate* a Tag shall overwrite its old key with the new key.	O, TACF	Tag	**By design**
562	6.3.2.12.3.13	If the Tag does not write the new key successfully then it shall revert to the prior stored key.	O, TACF	Tag	**By design**
563	6.3.2.12.3.13	An Interrogator shall prepend an unencapsulated *KeyUpdate* with a frame-sync (see 6.3.1.2.8).	O, TACF	Interrogator	**By design**
564	6.3.2.12.3.13	A Tag shall only accept a *KeyUpdate* after a successful cryptographic authentication.	O, TACF	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
565	6.3.2.12.3.13	Because an *Access* command sequence is not a cryptographic authentication, a Tag that most recently entered the **secured** state via a successful *Access* command sequence shall not execute a *KeyUpdate* and instead treat the command's parameters as unsupported (see Table C.30).	O, TACF	Tag	**By design**
566	6.3.2.12.3.13	A Tag shall reply to a *KeyUpdate* using the *in-process* reply specified in 6.3.1.6.3.	O, TACF	Tag	**By design**
567	6.3.2.12.3.13	The cryptographic suite shall specify the parameters that a Tag includes in its response.	O, TACF	Tag	**By design**
568	6.3.2.12.3.13	If a Tag receives a properly formatted *KeyUpdate* but there is a cryptographic error, and the cryptographic suite specifies that the error requires a security timeout, then the Tag shall set a security timeout as specified in 6.3.2.5.	O, TACF	Tag	**By design**
569	6.3.2.12.3.13	If a Tag that supports security timeouts for the *KeyUpdate* command receives a *KeyUpdate* during a timeout then it shall reject the command, backscatter an error code (see Annex I), and remain in its current state.	O, TACF	Tag	**By design**
570	6.3.2.12.3.14	Interrogators and Tags may implement the *TagPrivilege* command; if they do, they shall implement it as shown in Table 6.62.	O	Tag and Interrogator	**By demonstration** Issue a TagPrivilege command configured according a supported cryptographic suite. Verify that the Tag backscatters a valid response. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
571	6.3.2.12.3.14	A *TagPrivilege* contains 2 RFU bits. An Interrogator shall set these bits to 00_2.	O	Interrogator	**By design**
572	6.3.2.12.3.14	A Tag in the **secured** state that receives a *TagPrivilege* containing nonzero RFU bits shall not execute the *TagPrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
573	6.3.2.12.3.14	An unauthenticated Interrogator may issue a *TagPrivilege*; if it does then it shall issue the *TagPrivilege* without encapsulation and with Target=0 (i.e. specifying the access password).	O	Interrogator	**By design**
574	6.3.2.12.3.14	An authenticated Interrogator shall encapsulate a *TagPrivilege* in a *SecureComm* or *AuthComm* (see Table 6.28).	O	Interrogator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
575	6.3.2.12.3.14	If a Tag in the **secured** state receives an unencapsulated *TagPrivilege* from an authenticated Interrogator then it shall not execute the *TagPrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
576	6.3.2.12.3.14	A Tag in the **secured** state shall only read or modify the access-password privileges if the Interrogator supplied the correct access password and is not attempting to assert a deasserted privilege.	O	Tag	**By design**
577	6.3.2.12.3.14	In all other instances the Tag shall not execute the *TagPrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
578	6.3.2.12.3.14	A Tag in the **secured** state shall only read or modify a key's privileges if (a) the Interrogator authenticated itself as a crypto superuser and KeyID is assigned to the same cryptographic suite as that specified by CSI in the *Authenticate* command that preceded the *TagPrivilege*, or (b) KeyID is the same as that used by the Interrogator to authenticate itself and the Interrogator is not attempting to assert a deasserted privilege.	O	Tag	**By design**
579	6.3.2.12.3.14	If an Interrogator specifies Action=0 in a *TagPrivilege* then it may use any value for privilege. A Tag shall ignore privilege when Action=0.	O	Tag and Interrogator	**By design**
580	6.3.2.12.3.14	If Tag receives a *TagPrivilege* with Target=0 then it shall ignore the value that the Interrogator supplies for KeyID.	O	Tag	**By design**
581	6.3.2.12.3.14	Upon receiving an executable *TagPrivilege* with Action=1 a Tag shall overwrite the old privileges with the new privileges.	O	Tag	**By design**
582	6.3.2.12.3.14	If the Tag does not write the new privileges successfully then it shall revert to the prior stored privileges.	O	Tag	**By design**
583	6.3.2.12.3.14	A Tag in the **secured** state that receives a *TagPrivilege* which attempts to assert one or more RFU privilege bits or to change an unchangeable privilege value shall not execute the *TagPrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
584	6.3.2.12.3.14	An Interrogator shall prepend an unencapsulated *TagPrivilege* with a frame-sync (see 6.3.1.2.8).	O	Interrogator	**By design**
585	6.3.2.12.3.14	A Tag shall reply to a *TagPrivilege* using the *in-process* reply specified in 6.3.1.6.3.	O	Tag	**By design**
586	6.3.2.12.3.14	The Tag's response shall be as shown in Table 6.63 for Action=0 or Action=1.	O	Tag	**By design**
587	6.3.2.12.3.15	Interrogators and Tags may implement the *ReadBuffer* command; if they do, they shall implement it as shown in Table 6.64.	O	Tag and Interrogator	**By demonstration** Tested in Compliance with 6.3.1.6.4
588	6.3.2.12.3.15	BitCount specifies the number of bits to read. If BitCount=000_h then a Tag shall backscatter the contents of the ResponseBuffer starting at WordPtr and ending at the end of the allocated ResponseBuffer.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
589	6.3.2.12.3.15	A *ReadBuffer* contains 2 RFU bits. An Interrogator shall set these bits to 00_2.	O	Interrogator	**By design**
590	6.3.2.12.3.15	A Tag in the **open** or **secured** states that receives a *ReadBuffer* with nonzero RFU bits shall not execute the *ReadBuffer* and instead treat the command's parameters as unsupported (see Table C.30). Future protocols may use these RFU bits to expand the *ReadBuffer* command's functionality.	O	Tag	**By design**
591	6.3.2.12.3.15	An Interrogator may encapsulate a *ReadBuffer* in an *AuthComm* but shall not encapsulate it in a *SecureComm* (see Table 6.28).	O	Interrogator	**By design**
592	6.3.2.12.3.15	If a Tag implements a ResponseBuffer then that Tag shall implement the *ReadBuffer* command.	O	Tag	**By design**
593	6.3.2.12.3.15	An Interrogator shall prepend an unencapsulated *ReadBuffer* with a frame-sync (see 6.3.1.2.8).	O	Interrogator	**By design**
594	6.3.2.12.3.15	A Tag shall reply to a *ReadBuffer* using the *immediate* reply specified in 6.3.1.6.1.	O	Tag	**By design**
595	6.3.2.12.3.15	If **C**=1 and the memory bits specified in the *ReadBuffer* exist then the Tag's reply shall be as shown in Table 6.65 including a header (a 0-bit), the data bits, and the Tag's handle.	O	Tag	**By design**
596	6.3.2.12.3.15	The reply includes a CRC-16 calculated over the 0-bit, data bits, and handle. If one or more of the memory bits specified in the *ReadBuffer* do not exist, or if the **C** flag in XPC_W1 is zero-valued, then the Tag shall not execute the *ReadBuffer* and instead backscatter an error code (see Table C.30, unsupported parameters) within time T_1 in Table 6.16 rather than the reply shown in Table 6.65.	O	Tag	**By design**
597	6.3.2.12.3.16	Interrogators and Tags may implement the *Untraceable* command; if they do, they shall implement it as shown in Table 6.66.	O, EAS, TAC	Tag and Interrogator	**By demonstration** Issue an *Untraceable* command that is configured to hide the User memory. Verify that the *Read* command is only executed for an Interrogator that has an asserted Untraceable privilege. Issue another Untraceable command, which is configured to expose the User memory. Verify that the *Read* command is executed independent from the Untraceable privilege of the Interrogator. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
598	6.3.2.12.3.16	Upon receiving an *Untraceable* command a Tag that supports the **U** bit shall overwrite bit $21C_h$ of XPC_W1 with the provided U value regardless of the lock or permalock status of EPC memory.	O, EAS, TAC	Tag	**By design**
599	6.3.2.12.3.16	If the Tag does not support the **U** bit then the Tag shall ignore the provided U value but continue to process the remainder of the *Untraceable*.	O, EAS, TAC	Tag	**By design**
600	6.3.2.12.3.16	Upon receiving an *Untraceable* command a Tag shall overwrite its EPC length field (StoredPC bits 10_h – 14_h) with the provided length bits regardless of the lock or permalock status of EPC memory.	O, EAS, TAC	Tag	**By design**
601	6.3.2.12.3.16	A Tag shall execute a range change prior to replying to the *Untraceable*.	O, EAS, TAC	Tag	**By design**
602	6.3.2.12.3.16	If a Tag does not support range reduction then it shall ignore range but continue to process the remainder of the *Untraceable.*	O, EAS, TAC	Tag	**By design**
603	6.3.2.12.3.16	An Interrogator shall set these bits to 00_2.	O, EAS, TAC	Interrogator	**By design**
604	6.3.2.12.3.16	A Tag in the **secured** state that receives an *Untraceable* with nonzero RFU bits, TID=11_2, or range=11_2 shall not execute the *Untraceable* and instead treat the command's parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**
605	6.3.2.12.3.16	If a Tag in the **secured** state receives an *Untraceable* from an Interrogator with an asserted Untraceable privilege then it shall execute the command.	O, EAS, TAC	Tag	**By design**
606	6.3.2.12.3.16	If the Interrogator has a deasserted Untraceable privilege then the Tag shall not execute the command and instead treat the command's parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**
607	6.3.2.12.3.16	An authenticated Interrogator shall encapsulate an *Untraceable* in a *SecureComm* or *AuthComm* (see Table 6.28).	O, EAS, TAC	Interrogator	**By design**
608	6.3.2.12.3.16	If a Tag in the **secured** state receives an unencapsulated *Untraceable* from an authenticated Interrogator then it shall not execute the *Untraceable* and instead treat the command's parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**
609	6.3.2.12.3.16	*Untraceable* commands shall be atomic, meaning that a Tag, upon receiving an executable *Untraceable*, shall discard its prior memory and range settings and implement the new ones.	O, EAS, TAC	Tag	**By design**
610	6.3.2.12.3.16	A Tag that supports **XI**=1_2 shall not execute an *Untraceable* that specifies length bits greater than 11101_2 and shall instead treat the command's parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
611	6.3.2.12.3.16	Regardless of these absolute bounds on length, if an *Untraceable* specifies a length value that a Tag does not support then the Tag shall not execute the *Untraceable* and instead treat the command's parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**
612	6.3.2.12.3.16	A Tag that is operating with reduced range shall do so for all commands regardless of whether an Interrogator has an asserted or a deasserted Untraceable privilege.	O, EAS, TAC	Tag	**By design**
613	6.3.2.12.3.16	A Tag shall execute supported access commands that operate on untraceably hidden memory if the commanding Interrogator has an asserted Untraceable privilege, but shall not execute these commands if the Interrogator has a deasserted Untraceable privilege.	O, EAS, TAC	Tag	**By design**
614	6.3.2.12.3.16	In the latter case a Tag shall behave as though untraceably hidden memory does not exist and treat the commands' parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**
615	6.3.2.12.3.16	A Tag that is untraceably hiding EPC memory shall not include any of the untraceably hidden EPC memory bits when replying to an *ACK*.	O, EAS, TAC	Tag	**By design**
616	6.3.2.12.3.16	An Interrogator shall prepend an unencapsulated *Untraceable* with a frame-sync (see 6.3.1.2.8).	O, EAS, TAC	Interrogator	**By design**
617	6.3.2.12.3.16	The details of this irreversible untraceability, including whether a Tag with irreversibly hidden memory will still alter its operating range, and vice versa, shall be Tag-manufacturer defined.	O, EAS, TAC	Tag	**By design**
618	6.3.2.12.3.16	A Tag shall reply to an *Untraceable* using the *delayed* reply specified in 6.3.1.6.2. Upon receiving an executable *Untraceable* a Tag shall perform the specified actions.	O, EAS, TAC	Tag	**By design**
619	6.3.2.12.3.16	If a Tag receives an *Untraceable* whose fields it supports but nonetheless cannot execute, such as if the *Untraceable* instructs the Tag to expose an irreversibly hidden portion of Tag memory or the Interrogator has a deasserted Untraceable privilege, then the Tag shall not execute the *Untraceable* and instead treat the command's parameters as unsupported (see Table C.30).	O, EAS, TAC	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
620	6.3.2.12.3.17	Interrogators and Tags may implement the *FileOpen* command; if they do, they shall implement it as shown in Table 6.67.	O, CE	Tag and Interrogator	**By demonstration** Issue a FileOpen command configured according a supported cryptographic suite. Verify that the Tag backscatters a valid response. Test conditions: Temp: 23 +/– 3 ºC Freq: 860, 910, & 960 MHz Power: 0 dBm at Tag antenna Modulation: DSB-ASK Tari: 12.5 µs RTcal: 31.25 µs PW: 0.5 Tari Modulation depth: 90% Rise/fall time: < 0.33 Tari TRcal: 66,7 µs DR: 64/3 M: 3 TRext: 1
621	6.3.2.12.3.17	An Interrogator shall set these bits to 00_2.	O, CE	Interrogator	**By design**
622	6.3.2.12.3.17	A Tag in the **open** or **secured** states that receives a *FileOpen* with nonzero RFU bits or that specifies FileNum=1111111111_2 (RFU FileNum) shall not execute the *FileOpen* and instead treat the command's parameters as unsupported (see Table C.30).	O, CE	Tag	**By design**
623	6.3.2.12.3.17	An authenticated Interrogator shall encapsulate a *FileOpen* in a *SecureComm* or *AuthComm* (see Table 6.28).	O, CE	Interrogator	**By design**
624	6.3.2.12.3.17	If a Tag in the **secured** state receives an unencapsulated *FileOpen* from an authenticated Interrogator then it shall not execute the *FileOpen* and instead treat the command's parameters as unsupported (see Table C.30).	O, CE	Tag	**By design**
625	6.3.2.12.3.17	If an Interrogator or a Tag support File_N, N>0 then that Interrogator or Tag shall implement a *FileOpen*.	O, CE	Tag and Interrogator	**By design**
626	6.3.2.12.3.17	An Interrogator shall prepend an unencapsulated *FileOpen* with a frame-sync (see 6.3.1.2.8).	O, CE	Interrogator	**By design**
627	6.3.2.12.3.17	If a Tag supports the *FileOpen* command then it shall implement the file (**F**) indicator (see 6.3.2.1.3).	O, CE	Interrogator	
628	6.3.2.12.3.17	A Tag shall reply to a *FileOpen* using the *immediate* reply specified in 6.3.1.6.1.	O, CE	Tag	**By design**
629	6.3.2.12.3.17	If the Tag has an allocated file at FileNum then it shall close the currently open file, open the specified file, and reply as shown in Table 6.68.	O, CE	Tag	**By design**
630	6.3.2.12.3.17	LastFile indicates whether the just-opened file has the largest assigned FileNum; if a Tag has a FileNum larger than that of the just-opened file then it shall set LastFile to 0, otherwise it shall set LastFile to 1.	O, CE	Tag	**By design**
631	6.3.2.12.3.17	If a Tag receives a *FileOpen* specifying the currently open file then it shall leave the file open and reply as specified in Table 6.68.	O, CE	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
632	6.3.2.12.3.17	If a Tag receives a *FileOpen* but does not have an allocated file at FileNum, or if User memory is untraceably hidden and the Interrogator has a deasserted Untraceable privilege, or if the Tag is otherwise unable to execute the *FileOpen*, then the Tag shall not execute the *FileOpen* and instead treat the command's parameters as unsupported (see Table C.30), reverting to the currently open file (or to no file if the Tag doesn't have any allocated files or if User memory is untraceably hidden and the Interrogator has a deasserted Untraceable privilege).	O, CE	Tag	**By design**
633	6.3.2.12.3.18	Interrogators and Tags may implement the *FileList* command; if they do, they shall implement it as shown in Table 6.69.	O	Tag and Interrogator	**By design**
634	6.3.2.12.3.18	For example, if FileNum=4 and AddlFiles=2 then the Tag shall provide information for File_4 and for the next two higher-numbered files (which may be File_5 and File_6 if the Tag manufacturer assigned file numbers sequentially or may be other files if the numbering is not sequential).	O	Tag	**By design**
635	6.3.2.12.3.18	A *FileList* contains 2 RFU bits. An Interrogator shall set these bits to 00_2.	O	Interrogator	**By design**
636	6.3.2.12.3.18	A Tag in the **open** or **secured** states that receives a *FileList* with nonzero RFU bits or that specifies FileNum=1111111111_2 (RFU FileNum) shall not execute the *FileList* and instead treat the command's parameters as unsupported (see Table C.30). Future protocols may use these RFU bits to expand the *FileList* command's functionality.	O	Tag	**By design**
637	6.3.2.12.3.18	An authenticated Interrogator shall encapsulate a *FileList* in a *SecureComm* or *AuthComm* (see Table 6.28).	O	Interrogator	**By design**
638	6.3.2.12.3.18	If a Tag in the **secured** state receives an unencapsulated *FileList* from an authenticated Interrogator then it shall not execute the *FileList* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
639	6.3.2.12.3.18	An Interrogator shall not specify AddlFiles=FF_h.	O	Interrogator	**By design**
640	6.3.2.12.3.18	If a Tag receives a *FileList* with AddlFiles=FF_h then the Tag shall behave as though it had received a *FileList* with AddlFiles=FD_h.	O	Tag	**By design**
641	6.3.2.12.3.18	A Tag shall reply to a *FileList* using the *in-process* reply specified in 6.3.1.6.3.	O	Tag	**By design**
642	6.3.2.12.3.18	A Tag's response shall be as shown in Table 6.70 and includes a message for each file for which the Interrogator requested information.	O	Tag	**By design**
643	6.3.2.12.3.18	If a Tag is *static* then AvailFileSize shall be zero.	O	Tag	**By design**
644	6.3.2.12.3.18	If a Tag has more than 1022 blocks of free memory then AvailFileSize shall be 1111111111_2.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
645	6.3.2.12.3.18	If a Tag receives a *FileList* with an unsupported FileNum, or AddlFiles exceeds the number of files above FileNum, or User memory is untraceably hidden and the Interrogator has a deasserted Untraceable privilege, or the Tag is otherwise unable to execute the *FileList*, then the Tag shall not execute the *FileList* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
646	6.3.2.12.3.19	Interrogators and Tags may implement the *FilePrivilege* command; if they do, they shall implement it as shown in Table 6.72.	O	Tag and Interrogator	**By design**
647	6.3.2.12.3.19	An Interrogator shall set these bits to 00_2.	O	Interrogator	**By design**
648	6.3.2.12.3.19	A Tag in the **secured** state that receives a *FilePrivilege* with nonzero RFU bits shall not execute the *FilePrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
649	6.3.2.12.3.19	An authenticated Interrogator shall encapsulate a *FilePrivilege* in a *SecureComm* or *AuthComm* (see Table 6.28).	O	Interrogator	**By design**
650	6.3.2.12.3.19	If a Tag in the **secured** state receives an unencapsulated *FilePrivilege* from an authenticated Interrogator then it shall not execute the *FilePrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
651	6.3.2.12.3.19	A Tag shall execute a *TagPrivilege* according to Table 6.71 which specifies, for each Action value, the privilege assignment that the Tag makes (if any), the fields in the *FilePrivilege* that the Tag ignores, the required Tag or file privilege to perform the requested operation, and the reply that the Tag backscatters. An Interrogator may set an ignored field in a *FilePrivilege* to any value.	O	Tag	**By design**
652	6.3.2.12.3.19	Upon receiving an executable *FilePrivilege* with Action=001_2, 011_2, 101_2, or 111_2 a Tag shall overwrite the current file privilege(s) with the new privilege.	O	Tag	**By design**
653	6.3.2.12.3.19	If the Tag does not write the new privilege successfully then it shall revert to the prior stored privilege.	O	Tag	**By design**
654	6.3.2.12.3.19	An Interrogator shall prepend an unencapsulated *FilePrivilege* with a frame-sync (see 6.3.1.2.8).	O	Interrogator	**By design**
655	6.3.2.12.3.19	A Tag's response to the *FilePrivilege*, for incorporation into the *in-process* reply specified in 6.3.1.6.3, shall be as shown in Table 6.73.	O	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
656	6.3.2.12.3.19	If a Tag receives a *FilePrivilege* that it cannot execute because the access password or key the Interrogator supplied has insufficient privileges, or the *FilePrivilege* contains an unsupported KeyID, or privilege is an RFU value, or User memory is untraceably hidden and the Interrogator has a deasserted Untraceable privilege, or the Tag is otherwise unable to execute the *FilePrivilege*, then the Tag shall not execute the *FilePrivilege* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
657	6.3.2.12.3.20	Interrogators and Tags may implement the *FileSetup* command; if they do, they shall implement it as shown in Table 6.74.	O	Tag and Interrogator	**By design**
658	6.3.2.12.3.20	An Interrogator shall set these bits to 00_2.	O	Interrogator	**By design**
659	6.3.2.12.3.20	A Tag in the **secured** state that receives a *FileSetup* with nonzero RFU bits shall not execute the *FileSetup* and instead treat the command's parameters as unsupported (see Table C.30). Future protocols may use these RFU bits to expand the *FileSetup* command's functionality.	O	Tag	**By design**
660	6.3.2.12.3.20	A Tag shall only execute a *FileSetup* issued by an Interrogator with a file superuser privilege (see 6.3.2.11.3).	O	Tag	**By design**
661	6.3.2.12.3.20	An authenticated Interrogator shall encapsulate a *FileSetup* in a *SecureComm* or *AuthComm* (see Table 6.28).	O	Interrogator	**By design**
662	6.3.2.12.3.20	If a Tag in the **secured** state receives an unencapsulated *FileSetup* from an authenticated Interrogator then it shall not execute the *FileSetup* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
663	6.3.2.12.3.20	A *static* Tag that supports the *FileSetup* command shall permit an Interrogator with the file superuser privilege to modify a file's type but never its size.	O	Tag	**By design**
664	6.3.2.12.3.20	A *static* Tag shall write the FileType in a *FileSetup* as the file's new type and shall ignore FileSize. An Interrogator may set FileSize to any value when communicating with a *static* Tag.	O	Tag	**By design**
665	6.3.2.12.3.20	A *dynamic* Tag shall permit an Interrogator with the file superuser privilege to modify a file's type and size.	O	Tag	**By design**
666	6.3.2.12.3.20	When increasing a file's size a *dynamic* Tag shall only allocate "free" memory (i.e. memory not currently allocated to another file) to the resized file.	O	Tag	**By design**
667	6.3.2.12.3.20	Regardless of whether a Tag is *static* or *dynamic*, after executing a *FileSetup* a Tag's response shall include both FileType and FileSize (even if the Tag made no changes to either one). See Table 6.75.	O	Tag	**By design**
668	6.3.2.12.3.20	An Interrogator shall prepend an unencapsulated *FileSetup* with a frame-sync (see 6.3.1.2.8).	O	Interrogator	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
669	6.3.2.12.3.20	A Tag's response to the *FileSetup*, for incorporation into the *in-process* reply specified in 6.3.1.6.3, shall be as shown in Table 6.75.	O	Tag	**By design**
670	6.3.2.12.3.20	The response includes the FileNum, FileType, and FileSize. If a Tag receives a *FileSetup* that it cannot execute because the access password or key that the Interrogator most recently supplied does not have a file superuser privilege, or User memory is untraceably hidden and the Interrogator has a deasserted Untraceable privilege, or the Tag is otherwise unable to execute the *FileSetup*, then the Tag shall not execute the *FileSetup* and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
671	6.3.2.12.3.20	If a *dynamic* Tag is unable to execute the FileSize in the *FileSetup* command then it shall not execute any portion of the *FileSetup* (i.e. it shall not change the FileType) and instead treat the command's parameters as unsupported (see Table C.30).	O	Tag	**By design**
672	Annex A	Although a general EBV may contain blocks of varying lengths, Tags and Interrogators manufactured according to this specification shall use blocks of length 8 bits (EBV-8).	M	Tag and Interro-gator	**By design**
673	Annex A	Tags and Interrogators shall use the EBV-8 word format specified in Table A.1.	M	Tag and Interro-gator	**By design**
674	Annex B	State-transition tables B.1 to B.7 shall define a Tag's response to Interrogator commands.	M	Tag	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19
675	Table B.1	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field), (2) a command with a CRC error, or (3) an unsupported command.		--	Definition. Not verified.
676	Table B.2	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field), (2) a command with a CRC error, or (3) an unsupported command.		--	Definition. Not verified.
677	Table B.3	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field), (2) a command with a CRC error, or (3) an unsupported command.		--	Definition. Not verified.
678	Table B.4	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field), (2) a command with a CRC error, or (3) an unsupported command.		--	Definition. Not verified.

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
679	Table B.5	"Unsupported parameters" shall mean an access command with a correct handle and CRC and that is recognizable by the Tag but contains or specifies (1) a nonzero or incorrect RFU value, (2) an unsupported CSI; (3) an encapsulated command that is unsupported or disallowed, (4) an unsupported or incorrect memory bank, memory location, address range, or FileNum, (5) a hidden or locked memory bank or location, (6) an unsupported file or files, (7) a command that requires encapsulation but is nonetheless unencapsulated (see Table 6.28), (8) a *delayed* or *in-process* reply and the specified operation causes the Tag to encounter an error, (9) an operation for which the Interrogator has insufficient privileges, (10) an unsupported cryptographic parameter, or (11) other parameters not supported by the Tag.	M	--	Definition. Not verified.
680	Table B.5	"Incorrect handle" shall mean an access command with a correct CRC and that is recognizable by the Tag but has an incorrect handle. The cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its cryptographic engine upon receiving a security command with an incorrect handle.	M	--	Definition. Not verified.
681	Table B.5	"Improper" shall mean a command (except *Req_RN* or *Query*) that is recognizable by the Tag but is interspersed between successive *Kill* or *Access* commands in a password-based kill or access command sequence, respectively (see Figure 6.24 and Figure 6.26).	M	--	Definition. Not verified.
682	Table B.5	"Invalid" shall mean an erroneous command, an unsupported command, a command with invalid parameters, a command with a CRC error, a command (other than a *Query*) with a session parameter not matching that of the inventory round currently in progress, or any other command either not recognized or not executable by the Tag.		--	Definition. Not verified.

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
683	Table B.6	"Unsupported parameters" shall mean an access command with a correct handle and CRC and that is recognizable by the Tag but contains or specifies (1) a nonzero or incorrect RFU value, (2) an unsupported CSI; (3) an encapsulated command that is unsupported or disallowed, (4) an unsupported or incorrect memory bank, memory location, address range, lock payload, blockpermalock payload, KeyID, or FileNum, (5) a hidden or locked memory bank or location, (6) an unsupported file or files, (7) insufficient or unallocateable memory, (8) an unencrypted message that requires encryption, (9) a command that requires encapsulation but is nonetheless unencapsulated see Table 6.28), (10) a *delayed* or *in-process* reply and the specified operation causes the Tag to encounter an error, (11) an RFU privilege value, (12) an operation for which the Interrogator has insufficient privileges, (13) an unsupported cryptographic parameter, or (14) other parameters not supported by the Tag.		--	Definition. Not verified.
684	Table B.6	"Incorrect handle" shall mean an access command with a correct CRC and that is recognizable by the Tag but has an incorrect handle. The default next state is **secured**, but the cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its crypto engine and transition to the **open** state upon receiving a security command with an incorrect handle.		--	Definition. Not verified.
685	Table B.6	"Improper" shall mean a command (*except Req_RN or Query*) that is recognizable by the Tag but is interspersed between successive Kill or Access commands in a password-based kill or access command sequence, respectively (see Figure 6.24 and Figure 6.26).		--	Definition. Not verified.
686	Table B.6	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field or a *BlockWrite*/*BlockErase* with a zero-valued WordCount), (2) a command with a CRC error, (3) an unsupported command, or (4) a *Write* command for which the immediately preceding command was not a *Req_RN*. The default next state is **secured**, but the cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its cryptographic engine and transition to the **open** state upon receiving an invalid command.		--	Definition. Not verified.
687	Annex C	Command-response tables C.1 to C.30 shall define a Tag's response to Interrogator commands.	M	Tag	**By design** Also tested in compliance with 6.3.2.4, Figure 6.19

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
688	Table C.30	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field), (2) a command with a CRC error, or (3) an unsupported command.		--	Definition. Not verified.
689	Table C.30	"Unsupported parameters" shall mean an access command with a correct handle and CRC and that is recognizable by the Tag but contains or specifies (1) a nonzero or incorrect RFU value, (2) an unsupported CSI; (3) an encapsulated command that is unsupported or disallowed, (4) an unsupported or incorrect memory bank, memory location, address range, or FileNum, (5) a hidden or locked memory bank or location, (6) an unsupported file or files, (7) a command that requires encapsulation but is nonetheless unencapsulated (see Table 6.28), (8) a *delayed* or *in-process* reply and the specified operation causes the Tag to encounter an error, (9) an operation for which the Interrogator has insufficient privileges, (10) an unsupported cryptographic parameter, or (11) other parameters not supported by the Tag.		--	Definition. Not ve<rified.
690	Table C.30	"Unsupported parameters" shall mean an access command with a correct handle and CRC and that is recognizable by the Tag but contains or specifies (1) a nonzero or incorrect RFU value, (2) an unsupported CSI; (3) an encapsulated command that is unsupported or disallowed, (4) an unsupported or incorrect memory bank, memory location, address range, lock payload, blockpermalock payload, KeyID, or FileNum, (5) a hidden or locked memory bank or location, (6) an unsupported file or files, (7) insufficient or unallocateable memory, (8) an unencrypted message that requires encryption, (9) a command that requires encapsulation but is nonetheless unencapsulated (see Table 6.28), (10) a delayed or in-process reply and the specified operation causes the Tag to encounter an error, (11) an RFU privilege value, (12) an operation for which the Interrogator has insufficient privileges, (13) an unsupported cryptographic parameter, or (14) other parameters not supported by the Tag.		--	Definition. Not ve<rified.
691	Table C.30	"Incorrect handle" shall mean an access command with a correct CRC and that is recognizable by the Tag but has an incorrect handle. The cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its cryptographic engine upon receiving a security command with an incorrect handle.		--	Definition. Not ve<rified.

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
692	Table C.30	"Incorrect handle" shall mean an access command with a correct CRC and that is recognizable by the Tag but has an incorrect handle. The default next state is **secured**, but the cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its crypto engine and transition to the **open** state upon receiving a security command with an incorrect handle.		--	Definition. Not ve<rified.
693	Table C.30	"Improper" shall mean a command (*except Req_RN or Query*) that is recognizable by the Tag but is interspersed between successive Kill or Access commands in a password-based kill or access command sequence, respectively (see Figure 6.24 and Figure 6.26).		--	Definition. Not ve<rified.
694	Table C.30	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field or a *BlockWrite*/*BlockErase* with a zero-valued WordCount), (2) a command with a CRC error, (3) an unsupported command, or (4) a *Write* command for which the immediately preceding command was not a *Req_RN*. The cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its cryptographic engine upon receiving an invalid command.		--	Definition. Not ve<rified.
695	Table C.30	"Invalid" shall mean a command not recognizable by the Tag such as (1) an erroneous command (example: a command with an incorrect length field or a *BlockWrite*/*BlockErase* with a zero-valued WordCount), (2) a command with a CRC error, (3) an unsupported command, or (4) a *Write* command for which the immediately preceding command was not a *Req_RN*.. The default next state is **secured**, but the cryptographic suite indicated by CSI in the prior *Challenge* or *Authenticate* command may specify that a Tag reset its cryptographic engine and transition to the **open** state upon receiving an invalid command.		--	Definition. Not verified.
696	Annex G.1	When an Interrogator in a multiple- or dense-Interrogator environment instructs Tags to use subcarrier backscatter, the Interrogator shall adopt the channel plan found at *http://www.gs1.org/epcglobal/implementation* for the regulatory region in which it is operating.	M	Interro-gator	**By design**
697	Annex G.1	When an Interrogator in a multiple- or dense-Interrogator environment instructs Tags to use FM0 backscatter, the Interrogator shall adopt the channel plan in accordance with local regulations.	M	Interrogat or	**By design**
698	Annex G.1	Interrogator signaling (both modulated and CW) shall be centered in a channel with the frequency accuracy specified in 6.3.1.2.1 (as appropriate), unless local regulations specify tighter frequency accuracy, in which case the Interrogator shall meet the local regulations.	M	Interro-gator	**By design** Also tested in compliance with 6.3.1.2.1.

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
699	Annex G.1	Interrogator transmissions shall satisfy the dense-Interrogator transmit mask in 6.3.1.2.11 (as appropriate), unless local regulations specify tighter mask, in which case the Interrogator shall meet the local regulations.	O	Interro-gator	**By design** Also tested in compliance with 6.3.1.2.11.
700	Annex G.1	If an Interrogator uses SSB-ASK modulation, the transmit spectrum shall be centered in the channel during R=>T signaling, and the CW shall be centered in the channel during Tag backscatter.	O	Interro-gator	**By demonstration** (only for Interrogators that implement SSB modulation in dense-Interrogator environments). You found the Easter Egg - Please drop a note to rfid@cisc.at . Test conditions: Temp: 23 +/– 3 ºC Freq: At channel frequency closest to center of supported band. Power: Maximum Interrogator transmit power, as implemented. Modulation: SSB Tari: 25 µs Backscatter data rate: One or more of the dense-interrogator data rates specified in Annex G of the Protocol specification, as implemented. Other transmit parameters: As implemented Measurement equipment setting: - Resolution bandwidth: 1 kHz - Video bandwidth: Equal to the RBW - Sweep Time: AUTO - Span: 1 MHz - Trace mode: Max hold sufficient to capture all emissions - Detection mode: Averaging Modulation methode: - continous repeated inventory sequence (no tags present)
701	Annex I.1	If a Tag is required to backscatter an error code then the Tag shall use one of the error codes shown in Table I.2	M	Tag	**By design**
702	Annex I.1	If a Tag supports error-specific codes, then it shall use the error-specific codes shown in Table I.2.	M	Tag	**By design**
703	Annex I.1	If a Tag does not support error-specific codes, then it shall backscatter error code 00001111_2 (indicating a non-specific error) as shown in Table I.2.	M	Tag	**By design**
704	Annex I.1	A Tag shall backscatter error codes only from the **open** or **secured** states.	M	Tag	**By design**
705	Annex I.1	A Tag shall not backscatter an error code if it receives an invalid or improper access command, or an access command with an incorrect handle.	M	Tag	**By design**
706	Annex I.1	If an error is described by more than one error code, then the more specific error code shall take precedence and shall be the code that the Tag backscatters.	M	Tag	**By design**
707	Annex J.1	A Tag in the **arbitrate** state shall decrement its slot counter every time it receives a *QueryRep* command, transitioning to the **reply** state and backscattering an RN16 when its slot-counter value reaches 0000_h.	M	Tag	**By design**

Item	Protocol Subclause	Requirement	MO	Applies To	How Verified
708	Annex J.1	A Tag that returns to **arbitrate** with a slot-counter value of 0000_h shall decrement its slot-counter from 0000_h to $7FFF_h$ (i.e. the slot counter rolls over) at the next *QueryRep* with matching session.	M	Tag	**By design**
709	Annex L.5	If **T**=0 then the **XI** bit may be either (i) the logical OR of bits 210_h–217_h of XPC_W1 or (ii) the logical OR of bits 210_h–218_h of XPC_W1; the Tag manufacturer shall choose whether the Tag implements (i) or (ii).	M	Tag	**By design, selected option shall be specified in the test submission**
710	Annex N	NOTE: All commands listed in Table N.1 are covered by the values in column MO in this table		Tag	
711	Annex N	NOTE: All commands listed in Table N.1are covered by the values in column MO in this table		Interrogator	
712	Annex N	6.3.2.1.3 an E2h class identifier and an XTID (see 4.1) are manda-tory	EAS TAC TACC TACA TACF CE	Tag	**By demonstration** Tested in compliance with Item 128
713	Annex N	6.3.2.1.3 a nonzero XTID serialization field is mandatory	EAS TAC TACC TACA TACF CE	Tag	**By demonstration** Tested in compliance with Item 128
714	Annex N	6.3.2.1.2.5 SLI and K bits in XPC_W1 are mandatory	EAS TAC TACC TACA TACF CE	Tag	**By design**
715	Annex N	6.3.2.1.2.5 NR bit in XPC_W1 is mandatory	TAC TACC TACA TACF CE	Tag	**By design**
716	Annex N	6.3.2.1.2.5 H bit in XPC_W1 is mandatory	CE	Tag	**By design**
717	Annex N	6.3.2.11 a Tag shall implement the mechanisms in this protocol that prevent it from transitioning directly from the acknowledged to the secured state	EAS TAC TACC TACA TACF CE	Tag	**By design**
718	Annex N	6.3.2.5 Security timeout for the Access command with a timeout range as specified in Table 6.20 is mandatory	EAS TAC TACC TACA TACF CE	Tag	**By demonstration** Apply access sequence with password $FAFA_h$. Begin access sequence after 19.9 ms and verify that tag does not accept it. Apply access sequence with password $FAFA_h$. Begin access sequence after 200.1 ms and verify that tag does accept it.
719	Annex N	6.3.2.1.41 >32 bits User memory is mandatory	EAS	Tag	**By demonstration** Read 33 bit of user memory and verify that there is no error code
720	Annex N	6.3.2.11.3 At least 2 files are mandatory	CE	Tag	**By demonstration** Open and read one byte of the first two files and verify that there is no error code

7 Revision History

Date & Version Number	Section(s)	Change
April 2015 Issue 1	All	Original Document

A Annex A

A.1 Scope

This annex provides additional explanation of conformance items, testing parameters, and equipment badging. Some of the questions answered here are referenced in the How Verified column of in the Protocol Requirements table in* MERGEFORMAT *Section 6* while others relate generally to the conformance process.

The terms Reader and Interrogator are synonymous.

A.2 Q and A

Q1: How does a reader vendor specify R=>T and T=>R parameters to be tested?

A: The reader vendor specifies modulation type, PIE ratio, DR and mask type (dense, multi or single) for each Tari/Backscatter Data Rate (BDR)/encoding combination they wish to have tested in the Mode Table (Table A-1 is a sample completed table). BDR is defined as Backscatter Link Frequency (BLF) divided by M.

Field entry options are:

- Modulation types: DSB-ASK, SSB-ASK or PR-ASK
- PIE ratio: A value in the range 1.5:1 to 2:1, inclusive
- DR: 8 or 64/3
- Mask type: DI (Dense Interrogator), MI (Multi Interrogator), or SI (Single Interrogator)

The vendor enters up to six Tari's to be tested. If more than one Tari value is to be tested, the vendor must list their minimum and maximum Tari.

The same encoding/BDR values can appear more than once (note M=8, BDR=32 entries in the subcarrier and FDM DI categories in Table A-1). This is necessary if a different modulation type, PIE ratio, DR value, or mask type is to be tested for the same Tari/BDR/encoding values.

A "mode" is defined as a combination of Tari, modulation type, PIE, DR, BDR, mask type, and encoding (i.e. a particular entry in the Mode Table). For example, Table A-1 indicates six "modes" for testing.

Table A-1 Sample of completed Reader Mode Table

Backscatter Encoding	M	Backscatter Data Rate (kbps)	Tari (µs)					
			25	7.14	10			
FM0	1	160			PR/2:1/8			
	1	320		DS/1.5:1/8				
	1	640		DS/1.5:1/64				
Subcarrier	8	64	PR/2:1/64					
	8	32	PR/1.5:1/64					
FDM DI	8	32	PR/2:1/64					
TDM DI								

Key:

Dense interrogator mask met (necessary but not sufficient for DI certification)
Multiple interrogator mask met
Single interrogator mask met

DS	DSB-ASK
SS	SSB-ASK
PR	PR-ASK
X:1	PIE ratio
8	DR=8
64	DR=64/3

Table A-2 is a Mode Table template where VS indicates Vendor Selection, parameters to be chosen by the ven-dor. Limits such as >1, indicate parameter restrictions.

Table A-2 Reader Mode Table Template

Backscatter Encoding	M	Backscatter Data Rate (kbps)	Tari (µs)					
			VS max	VS min	VS	VS	VS	VS
FM0	1	VS						
	1	VS						
	1	VS						
	1	VS						
	1	VS						
	1	VS						
Subcarrier	VS >1	VS						
	VS >1	VS						
	VS >1	VS						
	VS >1	VS						
	VS >1	VS						
	VS >1	VS						
FDM DI	VS >2	VS						
	VS >2	VS						
TDM DI	VS	VS						
	VS	VS						

Parameters declared in the table are tested. The entries uniquely determine the expected RTcal and TRcal values. The test facility will derive test limits from these values.

Q2: The Mode Table is informative but contains an overwhelming amount of information. Is the HW certified or not?

A: EPCglobal will list a Conformance Badge for Readers and Tags. For Readers, the Conformance Badge will direct the viewer to the mode table for more detailed information. The following are examples of Reader and Tag badges.

Reader Conformance Badge		**Options**
Reader or Module	Reader	*Reader, Module*
Intended Operating Region	US	*Intended region of operation*
Frequency Range	902 – 928 MHz	*Band of operation*
Modulation Types	PR-ASK and DSB-ASK	*PR-ASK, DSB-ASK, SSB-ASK*
Tari's	7.14 µs, 25 µs, 10 µs	*Tested Tari's*
Backscatter Encoding Support	FM0, Miller Subcarrier	*FM0, Miller Subcarrier*
Frequency Scheme	FHSS	*FHSS, Agility, Fixed*
Temperature Range	-40°C to 65°C	*Product temperature range*
Environment	Dense & Multi Interrogator	*Dense, Multi Interrogator*
Dense Operation*	FDM and TDM	*N/A, FDM, TDM*
Optional Command Support	Access, BlockWrite	*Access, BlockWrite, BlockErase*

* TDM only Dense Interrogator operators should not be used in a dense FDM deployment.

7Tag Conformance Badge		**Options**
Frequency Range	860 – 960 MHz	*Band of operation*
Backscatter Modulation Type	ASK	*ASK, PSK*
Temperature Range	-40°C to 65°C	*Product temperature range*
Optional Command Support	Access, BlockWrite	*Access, BlockWrite, BlockErase*

Q3: What are the criteria for receiving a DI or MI certification? Does DI or MI certified mean the Reader meets the respective mask in all modes tested?

A: To receive DI certification the vendor shall declare himself an FDM (Frequency Division Multiplexed) and/or TDM (Time Division Multiplexed) operator. An FDM operator intends to participate in a cooperative frequency plan with other FDM operators such that Reader transmissions do not spectrally interfere with Tag backscatter signaling. A TDM operator intends to be on/off multiplexed such that two near-by Readers never transmit simul-taneously. TDM DI only Readers should not be operated in an FDM DI deployment.

Criteria for getting DI certification are as follows:

1a) If FDM then test with Tari of 25 µs, PR-ASK or SSB-ASK, using a subcarrier with M=4 or 8, and declare other mode parameters

1b) If TDM then test in declared mode

2) Pass DI mask in tested mode

3) Pass channelization test (frequency accuracy)

MI certification is granted in the Conformance Badge if the MI mask is met for any one of the modes in the Mode Table; that is, at least one entry contains a yellow mark.

The vendor may choose to be tested against the DI mask in any mode. If the mask is met, this will be indicated in the Mode Table by a green mark. A mark, in itself, is not sufficient to achieve DI certification in the Conformance Badge. The other DI criteria must also be met, and if they are, a green mark will be indicated in the FDM DI or TDM DI portion of the Mode Table.

DI or MI certified does not mean the Reader meets the respective mask for all modes tested.

Q4: Can a Reader get a MI (yellow) and DI (green) mask mark at the same Tari, modulation type, PIE ratio, DR, BDR, and encoding values in the Mode Table?

A: No, either a DI, MI, or SI mask mark is given for a particular set of these parameters. If a DI mask is met, it supercedes MI and SI, so DI credit is given (green mark). Likewise, MI supercedes SI. The vendor chooses DI, MI, or SI testing for a particular set of these parameters. The vendor has the option to move to a less stringent mask if they cannot meet the more stringent mask during test.

If at least one of the parameters listed above is unique this will appear as a separate entry in the Mode Table. Both DI and MI certification can be achieved if at least two modes are specified and DI mask testing is per-formed for one and MI performed for the other. SI is not listed in the Conformance Badge unless neither the DI nor MI criteria are met.

Q5: If table entries are optional, what is the incentive for a vendor to attempt certification in multiple modes? The more modes elected, the greater possibility for failure. A vendor can get credit in the Conformance Badge by passing in just one mode. At the other extreme, test time can become excessive for vendors wishing testing at a large number of Tari's. What are the minimal and maximal test requirements?

A: Vendors are required to test at least one mode at their minimum and maximum Tari. If a Reader only sup-ports one Tari, that Tari is tested and it is shown in the Conformance Badge. If a Reader supports two or more Tari's, testing must occur at minimum and maximum Tari (at least two modes). The vendor can choose to get tested at up to six Tari's, at as many modulation types, PIE ratios, DR's, BDR's, and encoding values as they wish. The Conformance Badge will indicate the Tari's tested, not to exceed six values.

Q6: Numerous interrogator by demonstration items in the conformance document specify testing "At centre frequency closest to centre of supported band". What exactly does this mean?

A: The Gen2 protocol accommodates Readers from any region that regulates UHF RFID between 860 and 960 MHz Multiple operational frequency bands must therefore be supported in conformance testing. For tests in which "At centre frequency closest to centre of supported band" is specified as a test condition, the vendor declares this frequency to the testing facility according to the following criteria:

a. If the Reader is to be certified for operation in North America and supports subcarrier signalling, then the channelization specified in Table G.1 must be supported and the Reader is tested at 915.25 MHz which is the supported channel frequency closest to the centre of the band.

b. If the Reader is to be certified for operation in North America and does not support subcarrier signalling, the Reader is tested at the channel frequency closest to the band centre that the Reader supports. The vendor declares that frequency. The vendor may support a sub-band of the FCC band.

c. If the Reader is to be certified for operation in a region other than North America, the Reader is tested at the channel frequency closest to the band centre that the Reader supports. The vendor declares that frequency. The vendor may support a sub-band of the regional band.

If a Reader supports multiple regions, certification is achieved by separately testing each band according to the above guidelines.

The centre frequency definition has significance for Multi-Interrogator spectral mask testing (6.3.1.2.11, Figure 6.6). The following clarifies the procedures for Multi-Interrogator testing:

a. If the Reader is to be certified for operation in North America and supports subcarrier signalling, the spectral mask requirement (Figure 6.6) is centred at a valid channel frequency (Table G.1) for purposes of compliance test. For the purposes of defining the testing mask, a channel is 500 kHz wide.

b. If the Reader is to be certified for operation in North America and does not support subcarrier signalling, the spectral mask is centred at the vendor declared frequency for purposes of compliance test. For the purposes of defining the test mask, a channel is a maximum of 500 kHz wide. The vendor declares the channel width.

c. If the Reader is to be certified for operation in a region other than North America, the spectral mask is centred at the vendor declared frequency for purposes of compliance test. For the purposes of defining the test mask, channel width is determined by local regulations and is 200 kHz for a CEPT-regulated region.

Q7: The conformance document (6.3.1.2.1) specifies that dense-interrogator testing can be limited to the mini-mum or maximum temperature at which the Reader supports (see Test Condition excerpt below). How does this statement affect me in conformance testing?

Test conditions:

Temp: max (–40, minimum supported temperature) and min (65, maximum supported temperature). If sup-ported temperature range exceeds –25 or 40 then testing will also be performed at –25 or 40 respectively. All temperatures are in ºC (all +/– 3 ºC)

A: The intent of this wording to provide a certification path for Readers rated for narrower or wider temperature ranges while preventing spectral pollution when they are operated outside their rated range.

The reader vendor declares their rated temperature range on the conformance application form and shows evidence in their by-design documentation that the rated temperature is specified in their product specification. The test facility tests over the declared range. If the vendor passes, the tested range is listed in the vendors Conformance Badge. It is the end users responsibility to deploy the Reader in an environment that does not exceed the tested limits.

For Readers with rated ranges beyond the –40 or 65 limits, testing shall also be performed at –40 or 65, respectively. For Readers with rated ranges between –25 and –40 or between 40 and 65, testing shall also be per-formed at –25 or 40, respectively.

Q8: For purposes of testing Reader power-up settling time, what defines the end of the settling time interval? The Ts and Ths settling time intervals are shown in Figures 6.3 and 6.5, respectively, in the Gen2 protocol specification.

A: The Ts and Ths intervals end when the envelope settles to within 5% of its 100% electric field strength level.

Q9: The conformance document (6.3.1.2.6 and 6.3.1.2.7) specifies that the Reader RF envelope shall rise and fall monotonically between the specified power limits. Measurement parameters are not specified, so it is feasible that a Reader can fail the monotonicity test due to measurement uncertainty. What is the test procedure that accounts for measurement uncertainty?

A: The test set recovers a time-sampled profile of the rising and falling ramp of the RF envelope. Within the regions that the monotonicity requirements apply, samples are compared to all previous samples. In the case of a falling ramp, the current sample must be less than the previous sample within the measurement tolerance of the test set. For example, if the test set power measurement error is ±2%, than the current sample cannot exceed any of the previous samples by more than 2%. The test facility shall establish the measurement accuracy of the test set.

Q10: Testing of Reader modulated RF envelope characteristics and symbol durations are specified in 6.3.1.2.3 and 6.3.1.2.5 of the conformance document. These parameters are determined based on A and B measurements as shown in Figure 6.2 of the Gen2 protocol specification. In test, how is A determined?

A: In 6.3.1.2.5 of the Gen2 protocol specification, A is referred to as the maximum amplitude of the RF envelope. In Figure 6.2, A is shown as the midpoint between the maximum and minimum ripple excursions. The ripple represents inter-symbol interference associated with the band-limiting of the transmit symbols. Inter-symbol interference can case the RF envelope to exceed the maximum amplitude of an un-modulated signal with the same power. For consistency with the Gen2 protocol specification, the value of A in 6.3.1.2.3 and 6.3.1.2.5 shall be determined by measuring the un-modulated envelope immediately preceding modulation from the first Reader command issued after the end of the settling interval following a power-up. The test facility shall determine the optimal measurement time to establish an accurate estimate of A.

B DUT technical information required for test submission

B.1 Introduction

This document describes the required input parameters for Gen2V2 Conformance tests, which shall be supplied for each DUT (tag or reader) additional to the following documentation:

The product vendor shall supply datasheets and other information useful for communication with the tags and operation of the tags.

The documentation shall unambiguously define the brand name, model number, firmware version, and hardware version of the DUT.

The product vendor shall mark each DUT with a clearly visible number. Furthermore, it shall provide a list containing DUT number linked to UII memory content and TID content for each tag.

B.2 Application Conformance

This section applies for both tag and interrogator.

The supplied DUT supports the following applications according Annex N of the Gen 2 V2.0.0:

☒	M	mandatory functionality
☐	EAS	EAS functionality
☐	TAC	Tag Alteration (Core) functionality
☐	TACC	Tag Alteration (Core + Challenge) functionality
☐	TACA	Tag Alteration (Core + Authenticate) functionality
☐	TACF	Tag Alteration (Core + Full) functionality
☐	CE	Consumer Electronics functionality

B.3 Command support

This section applies for both tag and interrogator.

The supplied DUT supports the following optional commands:

☐ Challenge

☐ Access

☐ BlockWrite

☐ BlockErase

☐ BlockPermalock

☐ Authenticate

☐ AuthComm

☐ SecureComm

☐ KeyUpdate

☐ TagPrivilege

☐ ReadBuffer

☐ Untraceable

☐ FileOpen

☐ FileList

☐ FilePrivilege

☐ FileSetup

B.4 Configuration of tags

This section applies for tags only:

The product vendor shall complete the form below and/or provide a documentation containing the requested information.

- Is the Kill password implemented? ☐ Y / ☐ N
- Is the Access password implemented? ☐ Y / ☐ N
- Please provide the size of each memory bank:
  - Reserved Memory: _____ bit
  - EPC Memory: _____ bit
  - TID Memory: _____ bit
  - User Memory: _____ bit
- How many files are contained in the User Memory? _____
- What is the Block size for each file? _____
- If more than one file is implemented, how is the numbering (*FileNum*) of the files implemented? _____
- Are there any assignments of irreversible untraceability of memory regions (6.3.2.12.3.16)
- If applicable, default Tag privilege associated with access password (Gen2v2 table 6.22) _____
- If applicable, default Tag privilege associated with each cryptographic suite's Key (Gen2v2 table 6.23) _____
- If applicable, default File privileges (Gen2v2 table 6.24 and 6.25) _____

The following information is only required if at least one Security access command is supported:

- Which Authentication is supported by the device?
  - ☐ Tag Authentication
  - ☐ Interrogator Authentication
  - ☐ Mutual Authentication
- Which ISO/IEC 29167 Cryptographic Suite is supported?
  - ☐ Part ___
  - ☐ Part ___
  - ☐ More than 2 parts: _____
- How many keys are available? _____
- Keys shall be set as follows, whereas the leading part shall be used according the key length:
  - Key 0: 0x003A 0123 4567 8901 23…
  - Key 1: 0x015C 0123 4567 8901 23…
  - Key 2: 0x023A 0123 4567 8901 23…
  - Key 3: 0x035C 0123 4567 8901 23…
  - Key nn (for even nn): 0xnn3A 0123 4567 8901 23…
  - Key nn (for odd nn): 0xnn5C 0123 4567 8901 23…
- Please specify the implementation of the key management in case necessary for testing

- If multiple Cryptographic Suites are supported, please provide information on the assignment of the available Keys to the different Cryptographic Suites.

  _____

- What is the maximum size of the ResponseBuffer? _____
- What is maximal number of backscatter transmission in case of an In-process Tag reply? _____

B.5 Configuration of Interrogators

This section applies for interrogators only:

The following information is only required if at least one Security access command is supported:

- Which Authentication is supported by the device?
  - ☐ Tag Authentication
  - ☐ Interrogator Authentication
  - ☐ Mutual Authentication
- Which ISO/IEC 29167 Cryptographic Suite is supported?
  - ☐ Part ___
  - ☐ Part ___
  - ☐ More than 2 parts: _____