INTERNATIONAL STANDARD

ISO/IEC 29167-10

First edition
2015-05-15

Information technology — Automatic identification and data capture techniques —

Part 10: Crypto suite AES-128 security services for air interface communications

Technologies de l'information — Techniques automatiques d'identification et de capture de données —

Partie 10: Services de sécurité par suite cryptographique AES-128 pour communications par interface radio

Reference number
ISO/IEC 29167-10:2015(E)

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

Contents

Page

Foreword

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

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

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

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 31, *Automatic identification and data capture techniques*.

ISO/IEC 29167 consists of the following parts, under the general title Information technology — Automatic identification and data capture techniques:

— *Part 1: Security services for RFID air interfaces*

— *Part 10: Crypto suite AES-128 security services for air interface communications*

— *Part 11: Crypto suite PRESENT-80 security services for air interface communications*

— *Part 12: Crypto suite ECC-DH security services for air interface communication*

— *Part 13: Crypto suite Grain-128A security services for air interface communications*

— *Part 14: Crypto suite AES OFB security services for air interface communications*

— *Part 16: Crypto suite ECDSA-ECDH security services for air interface communications*

— *Part 17: Crypto suite cryptoGPS security services for air interface communications*

— *Part 19: Crypto suite RAMON security services for air interface communications*

The following parts are under preparation:

— *Part 15: Crypto suite XOR security services for air interface communications*

Introduction

This part of ISO/IEC 29167 specifies the security services of an AES-128 crypto suite for Tag authentication. AES has a fixed block size of 128 bits and a key size of 128 bits, 192 bits, or 256 bits. The version specified in this crypto suite uses AES with a fixed key size of 128 bits and is referred to as AES-128.

This part of ISO/IEC 29167 defines procedures for Tag Authentication using AES-128 and provides the following functionality:

— Tag Authentication;

— Tag Authentication allows authenticated reading of a part of the Tag's memory;

— Authenticated reading might be in plain text, MAC protected, Encrypted, or Encrypted and MAC protected;

— Crypto suite uses encryption for enciphering of plain text, as well as deciphering of encrypted text.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document might involve the use of patents concerning radio-frequency identification technology given in the clauses identified below.

ISO and IEC take no position concerning the evidence, validity, and scope of these patent rights.

The holders of these patent rights have assured the ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC. Information on the declared patents can be obtained from:

Impinj, Inc.
701 N 34th Street, Suite 300 **Seattle, WA 98103 USA**

The latest information on IP that might be applicable to this part of ISO/IEC 29167 can be found at www.iso.org/patents.

Information technology — Automatic identification and data capture techniques —

Part 10: Crypto suite AES-128 security services for air interface communications

1 Scope

This part of ISO/IEC 29167 defines the crypto suite for AES 128 for the ISO/IEC 18000 air interfaces standards for radio frequency identification (RFID) devices. Its purpose is to provide a common crypto suite for security for RFID devices that might be referred by ISO committees for air interface standards and application standards.

This part of ISO/IEC 29167 specifies a crypto suite for AES 128 for air interface for RFID systems. The crypto suite is defined in alignment with existing air interfaces.

This part of ISO/IEC 29167 defines various authentication methods and methods of use for the cipher. A Tag and an Interrogator can support one, a subset, or all of the specified options, clearly stating what is supported.

2 Conformance

2.1 Air interface protocol specific information

To claim conformance with this part of ISO/IEC 29167, an Interrogator or Tag shall comply with all relevant clauses of this part of ISO/IEC 29167, except those marked as “optional”.

2.2 Interrogator conformance and obligations

To conform to this part of ISO/IEC 29167, an Interrogator shall

— implement the mandatory commands defined in this part of ISO/IEC 29167 and conform to the relevant part of ISO/IEC 18000.

To conform to this part of ISO/IEC 29167, an Interrogator can

— implement any subset of the optional commands defined in this part of ISO/IEC 29167.

To conform to this part of ISO/IEC 29167, the Interrogator shall not

— implement any command that conflicts with this part of ISO/IEC 29167, or

— require the use of an optional, proprietary, or custom command to meet the requirements of this part of ISO/IEC 29167.

2.3 Tag conformance and obligations

To conform to this part of ISO/IEC 29167, a Tag shall

— implement the mandatory commands defined in this part of ISO/IEC 29167 for the supported types and conform to the relevant part of ISO/IEC 18000.

To conform to this part of ISO/IEC 29167, a Tag can

— implement any subset of the optional commands defined in this part of ISO/IEC 29167.

To conform to this part of ISO/IEC 29167, a Tag shall not

— implement any command that conflicts with this part of ISO/IEC 29167, or

— require the use of an optional, proprietary, or custom command to meet the requirements of this part of ISO/IEC 29167.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies

ISO/IEC 18000-63, *Information technology — Radio frequency identification for item management — Part 63: Parameters for air interface communications at 860 MHz to 960 MHz Type C*

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

ISO/IEC 29167-1, *Information technology — Automatic identification and data capture techniques — Part 1: Security services for RFID air interfaces*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762 (all parts) and the following apply.

4.1
AES-CBC-ENC(IV, key, data)
AES data encryption (forward operation) in CBC mode of input data "data", using initialization vector IV and 128 bit cryptographic key "key"

4.2
AES-ECB-ENC(key, data)
AES data encryption (forward operation) in ECB mode of input data "data", using 128 bit cryptographic key "key"

4.3
AES-CMAC-96(key, data)
CMAC generation using AES in forward operation with 128 bit cryptographic key "key" of input data "data", truncating the result by using only the 96 most significant bits from the 128-bit CMAC code

4.4
bit string
ordered sequence of 0's and 1's

4.5
block cipher
family of functions and their inverse functions that is parameterized by cryptographic keys; the functions map bit strings of a fixed length to bit strings of the same length

4.6
block size
number of bits in an input (or output) block of the block cipher

4.7
cryptographic key
string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa or to produce a message authentication code

4.8
CMAC
cipher-based MAC algorithm based on a symmetric key block cipher

Note 1 to entry: See MAC method 5 in Reference [1] for a normative reference.

4.9
Command (Message)
data that Interrogator sends to Tag with "Message" as parameter

4.10
D
number of additional 128-bit blocks with custom data that may be added to the Tag authentication response

4.11
Data Block (Block)
sequence of bits whose length is the block size of the block cipher

4.12
initialization vector
data block that some modes of operation require as an additional initial input

4.13
input block
data that is an input to either the forward cipher function or the inverse cipher function of the block cipher algorithm

4.14
Key
string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa or to produce a message authentication code

4.15
KeyID
numerical designator for a single key

4.16
Key[KeyID].ENC_key
key that shall be used for encryption

4.17
Key[KeyID].MAC_key
key that may be used for cryptographic integrity protection

4.18
MAC_key
Variable that shall contain the key that will be used for cryptographic integrity protection

4.19
Memory Profile
start pointer within the Tag's memory for addressing custom data block

4.20
Message
part of the Command that is defined by the crypto suite

4.21
Mode of Operation (Mode)
algorithm for the cryptographic transformation of data that features a symmetric key block cipher algorithm

4.22
output block
data that is an output of either the forward cipher function or the inverse cipher function of the block cipher algorithm

4.23
Plaintext
usable data that is formatted as input to a mode

4.24
Reply (Response)
data that Tag returns to the Interrogator with "Response" as parameter

4.25
Response
part of the Reply (stored or sent) that is defined by the crypto suite

4.26
word
bit string comprised of 16 bits

5 Symbols and abbreviated terms

5.1 Symbols

xxxxb	binary notation of term "xxxx", where "x" represents a binary digit.
xxxxh	hexadecimal notation of term "xxxx", where "x" represents a hexadecimal digit. In this crypto suite the bytes in the hexadecimal numbers are presented with the most significant byte at the left and the least significant byte at the right. The bit order per byte is also presented with the most significant bit at the left and the least significant bit at the right. For example in "ABCDEF" the byte "AB" is the most significant byte and the byte "EF" is the least significant byte.
||	concatenation of syntax elements, transmitted in the order written (from left to right). For example "123456" || "ABCDEF" results in "123456ABCDEF", where the byte "12" is the most significant byte and the byte "EF" is the least significant byte.
Field[a:b]	Selection from a string of bits in Field. For a > b, selection of a string of bits from the bit string Field. Selection ranges from bit number a until and including bit number b from the bits of the string in Field, whereby Field[0] represents the least significant bit. For example Field[2:0] represents the selection of the three least significant bits of Field.

5.2 Abbreviated terms

AES	Advanced Encryption Standard
CBC	Cipher-Block Chaining
CMAC	Cipher-based MAC
ECB	Electronic Code Book
FIPS	Federal Information Processing Standard

IV	Initialization Vector
LSB	Least Significant Byte
MAC	Message Authentication Code
MPI	Memory Profile Indicator
MSB	Most Significant Byte
NIST	(United States) National Institute of Standards and Technology
RFU	Reserved for Future Use
TID	Tag-IDentification or Tag IDentifier, depending on context
UII	Unique Identification ID

6 Introduction of the AES-128 crypto suite

The Advanced Encryption Standard (AES) is an open, royalty-free, symmetric block cipher based on so-called substitution-permutation networks. AES is highly suitable for efficient implementation in both software and hardware, including extremely constrained environments such as RFID Tags. The AES cipher is standardized as ISO/IEC 18033-3.[2]

AES is approved by the National Institute of Standards and Technology (NIST). It was approved as a standard in 2001 following a five-year standardization process that involved a number of competing encryption algorithms and published as FIPS PUB 197 in November 2001.

AES was originally published, along with design criteria and test vectors, in reference document [5] in the Bibliography.

NOTE AES normally uses encryption for the enciphering of plain text and decryption for the deciphering of encrypted text. This crypto suite uses encryption for enciphering of plain text as well as deciphering of encrypted text. This allows the use of an encryption-only implementation on the Tag.

7 Parameter definitions

Table 1 describes all the parameters that are used in this part of ISO/IEC 29167.

Table 1 — Definition of AES-128 crypto suite parameters

Parameter	Description
C_TAM1[15:0]	16-bit predefined constant for TAM1 with the value “96C5h” (for Tag to Interrogator response)
C_TAM2[15:0]	16-bit predefined constant for TAM2 with the value “96C5h” (for Tag to Interrogator response)
Ciphertext[n]	Temporary storage for encryption result
CUSTOMDATA(D*128)	Part of the Tag’s memory that may be returned with the Tag authentication response
IChallenge_TAM1[79:0]	80-bit challenge that the Interrogator generates for use in TAM1
IChallenge_TAM2[79:0]	80-bit challenge that the Interrogator generates for use in TAM2
Key[KeyID]	Keyset identified by KeyID, consisting of ENC_key for encryption and (optional) MAC_key for integrity protection
MAC_key[127:0]	Variable that shall contain the key that will be used for cryptographic integrity protection

Table 1 *(continued)*

Parameter	Description
TRnd_TAM1[31:0]	32-bit random data provided by the Tag for TAM1
TRnd_TAM2[31:0]	32-bit random data provided by the Tag for TAM2

8 Crypto suite state diagram

After power-up or reset the crypto suite transitions to its **Initial** state.

A transition to **Initial** state shall also cause a reset of all variables used by the crypto suite.

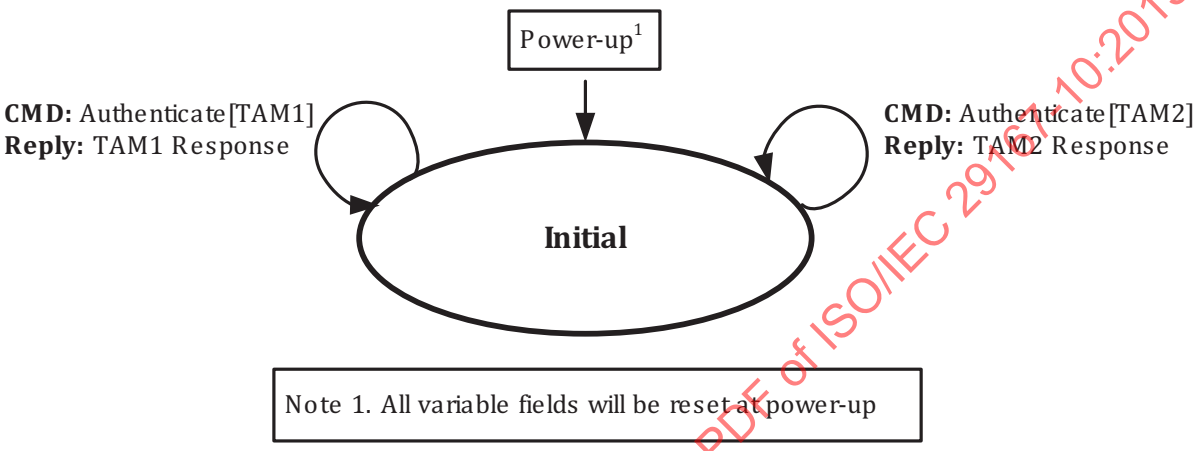

Figure 1 — Crypto suite Tag state diagram

9 Initialization and resetting

After power-up and after a reset the crypto suite transitions into the **Initial** state.

Implementations of this crypto suite shall assure that all memory used for intermediate results is cleared after each operation (message-response pair) and after reset.

10 Authentication

10.1 Introduction

This part of ISO/IEC 29167 supports only Tag Authentication. All functions are implemented using a message-response exchange. This section describes the details of the messages and responses that are exchanged between the Interrogator and Tag.

All message and response exchanges are listed in Table 2.

Table 2 — message and response functions

Command	Function
TAM1 message	Send Interrogator challenge and request Tag authentication response
TAM1 response	Return Tag authentication response
TAM2 message	Send Interrogator challenge and request Tag authentication response plus custom data
TAM2 response	Return Tag authentication response and custom data

10.2 Message and Response formatting

Message and Response are part of the security commands that are described in the air interface specification. The "air interface part" of the Tag passes the Message on to the "crypto suite part" of the Tag and returns the Response from the "crypto suite part" to the Interrogator. The crypto suite shall parse the Messages and process the data based on the value of AuthMethod, which is the first parameter of all Messages.

The following sections of this document describe the formatting of Message and Response for Tag Authentication. AuthMethod shall be "00_b" for Tag Authentication.

If AuthMethod = "00_b" the Tag shall parse Message as described in 10.3

If AuthMethod = "01_b", "10_b" or "11_b" then the Tag shall return a "Not Supported" error condition and shall transition to the **Initial** state.

10.3 Tag authentication (Method "00" = TAM)

10.3.1 TAM1 and TAM2

Tag Authentication allows an Interrogator to authenticate a Tag by verifying the Tag's secret key (TAM1). Optionally the Tag may return part of its memory as custom data, that may be protected (protection of integrity, authenticity of origin, and timeliness) and/or encrypted (confidentiality protection), with the Tag authentication TAM2 response.

The functionality shall be implemented by means of a challenge-response exchange. Tag authentication only shall be implemented in TAM1 and Tag authentication with the addition of custom data shall be implemented as TAM2 (see Figure 2).

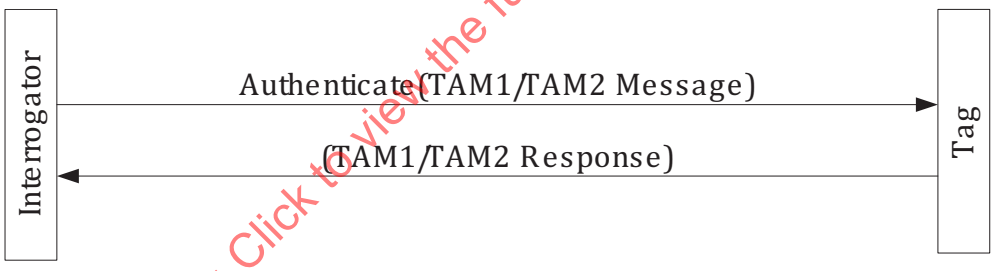

Figure 2 — Tag authentication

The following sections of this document describe the formatting of Message and Response for Tag Authentication.

The crypto suite shall parse the TAM Messages and process the data based on the value of CustomData, which is the second parameter of both TAM Messages. The Messages for Tag Authentication without and with custom data shall be distinguished by CustomData. CustomData shall be "0_b" for Tag Authentication without custom data and "1_b" for Tag Authentication with custom data.

If CustomData = "0_b" the Tag shall parse the TAM1 Message as described in 10.3.2

If CustomData = "1_b" the Tag shall parse the TAM2 Message as described in 10.3.5

10.3.2 TAM1 Message

For Tag authentication the Interrogator shall generate an 80-bit random TAM1 Interrogator challenge and include that in the TAM1 message. The TAM1 message shall also include the reference KeyID to select an encryption key in the Key Management Table (see Clause 12).

The TAM1 Message format has the following fields:

— AuthMethod: value "00_b" defines the use for TAM

— CustomData: flag to indicate that custom data is requested, "0_b" defines no custom data requested (TAM1)

— TAM1_RFU: makes the total length of the TAM1 Message a multiple of 8-bits and will be used for future extensions of this part of ISO/IEC 29167

— KeyID: defines the key that shall be used for TAM1

— IChallenge_TAM1: random challenge that the Interrogator has generated for use in TAM1

Table 3 — TAM1 Message format

	AuthMethod	CustomData	TAM1_RFU	KeyID	IChallenge_TAM1
# of bits	2	1	5	8	80
description	00_b	0_b	00000_b	[7:0]	random Interrogator challenge

The Tag shall accept this message in any state. If the parameters of the message are valid, then the Tag shall transition to the **Initial** state; thereby aborting any cryptographic protocol that has not yet been completed.

If the length of the TAM1 message <> 96 bits then the Tag shall return an "Other Error" error condition and shall transition to the **Initial** state.

If TAM1_RFU <> "00000_b" then the Tag shall return a "Not Supported" error condition and shall transition to the **Initial** state.

If the Tag does not support key[KeyID].ENC_key then it shall return a "Not Supported" error condition and shall transition to the **Initial** state.

10.3.3 TAM1 Response

If all verifications are successful then the Tag shall generate the random data TRnd_TAM1 (32 bits) and encrypt the challenge IChallenge_TAM1 of the Interrogator using Key[KeyID].ENC_key, after first prefixing the constant C_TAM1 (16 bits) and the random data TRnd_TAM1.

Table 4 — Response if optional fields have not been used

	TResponse
# of bits	128
description	AES-ECB-ENC(Key[KeyID].ENC_key, C_TAM1[15:0] || TRnd_TAM1[31:0] || IChallenge_TAM1[79:0])

After returning the TAM1 Response (TResponse) the Tag shall remain in the **Initial** state.

10.3.4 Final Interrogator processing TAM1

The Interrogator (or the external application controlling the Interrogator) decrypts the TAM1 Response (TResponse) and shall verify whether: C_TAM1 and IChallenge_TAM1 have the correct value. If the values are correct, then the Tag can be considered as authentic.

10.3.5 TAM2 Message

TAM2 shall be used for Tag Authentication if the Tag needs to return part of its memory as custom data that may be protected (protection of integrity and authenticity) and/or encrypted (confidentiality protection) with the Tag authentication. The TAM2 message shall also include the reference KeyID to

select an encryption key in the Key Management Table (see Clause 12). If protection of integrity and authenticity of the data is requested the selected key shall also contain a MAC key.

A Tag that supports TAM2 shall define at least one and at most 16 memory profiles. All supported addresses or pointers for the memory profiles shall be specified in Annex E of this part of ISO/IEC 29167.

The memory profiles may also be linked to a key in the Key Management Table that shall be used for the encryption process to protect the data.

Custom data is specified as a number (1 to 16) of consecutive 64-bit blocks in the Tag's memory. The custom data block shall be defined by the parameters Profile, Offset and BlockCount.

Profile shall select one of the memory profiles that are supported by the Tag.

Offset specifies a 12-bit offset (in multiples of 64-bit blocks) that needs to be added to the address that is specified by Profile. Minimum value "000000000000_b" corresponds to a zero offset. Maximum value is 111111111111_b (decimal 4095) corresponds to the maximum bit pointer offset of decimal value 262080.

BlockCount specifies the 4-bit number of 64-bit custom data blocks that need to be returned from the offset position onwards. Minimum value is "0000_b", corresponding to one single 64-bit block. Maximum binary value is "1111_b", or decimal 15, corresponds to a maximum number of 16 64-bit blocks of custom data that shall be returned. If the number of returned bits of the custom data is not a multiple of 128 then padding with zeroes shall be applied to the least significant bits of the last block that has a non-zero block size of less than 128 bits. The Interrogator shall maintain the value of BlockCount for use as part of the MAC verification process. The Tag manufacturer shall specify the number of custom data blocks that can be returned.

NOTE Access rights to custom data may be restricted by the specification of the interface. Annex E describes protocol specific implementations for various modes of the ISO/IEC 18000 international standards.

ProtMode specifies the mode of operation that shall be used for the encipherment and/or protection of the custom data. Table 5 defines the mode of operation for encipherment algorithms and/or message authentication algorithms for the (optional) protection (authentication and/or encipherment) of custom data.

Table 5 — Supported modes of operation for ProtMode

ProtMode[3:0] [a]	Description
0000_b	Plaintext (no integrity and/or confidentiality protection requested)
0001_b	CBC (encipherment only)
0010_b	CMAC (message authentication only)
0011_b	CBC + CMAC
0100_b	Reserved for Future Use
....	....
1111_b	Reserved for Future Use

[a] When a ProtMode is selected that specifies data encipherment (ProtMode "0001_b" and "0011_b") the Tag may assert a privilege that makes all memory accessible for the duration of the execution of the command. See Annex E for details of the air interface specific implementations.

Tags shall implement at least one of the modes of operation as defined by Table 5 for each of the memory profiles that the Tag supports.

The Interrogator shall generate an 80-bit random TAM2 Interrogator challenge in the following TAM2 message and include several fields indicating additional options for the authentication protocol.

The TAM2 Message format has the following fields:

— AuthMethod: value "00_b" defines the use for TAM

— CustomData: flag to indicate that custom data is requested, "1_b" defines custom data requested (TAM2)

— TAM2_RFU: makes the total length of the TAM2 Message a multiple of 8-bits and will be used for future extensions of this part of ISO/IEC 29167

— Profile: 4-bit pointer that selects a memory profile for the addition of custom data

— Offset: 12-bit value that specifies the number of multiples of 64-bit blocks that needs to be added to the address that is specified by Profile to define the first address of the custom data block.

— BlockCount: 4-bit number to define the size of the customer data as a number of 64-bit blocks

— ProtMode: 4-bit value to select the mode of operation that shall be used to process the custom data

— KeyID: defines the key that will be used for TAM2

— IChallenge_TAM2: random challenge that the Interrogator has generated for use in TAM2

Table 6 — TAM2 Message format

	Auth Method	Custom-Data	TAM2_ RFU	KeyID	IChallenge _TAM2	Profile	Offset	Block Count	ProtMode
# of bits	2	1	5	8	80	4	12	4	4
Description	00_b	1_b	00000_b	[7:0]	random Interrogator challenge	[3:0]	[11:0]	[3:0]	[3:0]

The Tag shall accept this message in any state. If the parameters of the message are valid, then the Tag shall transition to the **Initial** state; thereby aborting any cryptographic protocol that has not yet been completed.

If the length of the TAM2 message <> 120 bits then the Tag shall return an "Other Error" error condition and shall transition to the **Initial** state.

If TAM2_RFU <> "00000_b" then the Tag shall return a "Not Supported" error condition and shall transition to the **Initial** state.

If the Tag does not support key[KeyID].ENC_key then it shall return a "Not Supported" error condition and shall transition to the **Initial** state.

The Tag shall check if the specified memory profile has the right to use KeyID for further processing:

If Profile = "0000_b" and key[KeyID].MPI[0:0] = "1_b" or

If Profile = "0001_b" and key[KeyID].MPI[1:1] = "1_b" or

if Profile = "0010_b" and key[KeyID].MPI[2:2] = "1_b" or

if Profile = "0011_b" and key[KeyID].MPI[3:3] = "1_b" or

if Profile = "0100_b" and key[KeyID].MPI[4:4] = "1_b" or

if Profile = "0101_b" and key[KeyID].MPI[5:5] = "1_b" or

if Profile = "0110_b" and key[KeyID].MPI[6:6] = "1_b" or

if Profile = "0111_b" and key[KeyID].MPI[7:7] = "1_b" or

if Profile = "1000_b" and key[KeyID].MPI[8:8] = "1_b" or

if Profile = "1001_b" and key[KeyID].MPI[9:9] = "1_b" or

if Profile = "1010_b" and key[KeyID].MPI[10:10] = "1_b" or

if Profile = "1011_b" and key[KeyID].MPI[11:11] = "1_b" or

if Profile = "1100_b" and key[KeyID].MPI[12:12] = "1_b" or

if Profile = "1101_b" and key[KeyID].MPI[13:13] = "1_b" or

if Profile = "1110_b" and key[KeyID].MPI[14:14] = "1_b" or

if Profile = "1111_b" and key[KeyID].MPI[15:15] = "1_b",

then key[KeyID] is authorized for this memory profile, else key[KeyID] is not authorized for this memory profile and the Tag shall return a "Not Supported" error condition and shall transition to the **Initial** state.

If the memory profile specified in Profile is not supported by the Tag then the Tag shall return a "Not Supported" error condition.

If the block of custom data specified by Profile, Offset and BlockCount is not supported by the Tag then the Tag shall return a "Memory Overrun" error condition.

The Tag shall verify if the value of ProtMode is "0000_b" or "0001_b" or "0010_b" or "0011_b". If ProtMode has any other value the Tag shall return a "Not Supported" error condition and shall transition to the **Initial** state.

10.3.6 TAM2 Response

If all verifications are successful then the Tag shall proceed with parsing the TAM2 message.

The Tag encrypts the challenge IChallenge_TAM2 (80 bits) of the Interrogator after first prefixing the constant C_TAM2 (16 bits), the random data TRnd_TAM2 (32 bits) and possibly additional custom data as specified by Profile, Offset and BlockCount. **D** represents the number of additional 128-bit custom data blocks and is defined by BlockCount. If **n** is the decimal representation of BlockCount (0–15), then **D** := (**n+1**) DIV 2 + (**n+1**) MOD 2. CUSTOMDATA(**D***128) refers to the binary data string (including padding) with a length of **D***128 bits representing the selected custom data in plaintext format.

NOTE By design, the minimum value of **D** is 1. The maximum value of **D** supported by the Tag is specified by the Tag manufacturer.

10.3.6.1 Response if ProtMode = "0000_b": Plaintext

Custom data is added in plaintext. The initialization vector IV for the encryption shall contain all zeroes.

KeyID identifies the encryption key for CBC encipherment (with AES in forward operation).

Table 7 — Response if ProtMode = "0000_b": Plain text

	TResponse
# of bits	(1+**D**)*128
Description	AES-CBC-ENC(IV=0, Key[KeyID].ENC_key, C_TAM2[15:0] || TRnd_TAM2[31:0] || IChallenge_TAM2[79:0]) || CUSTOMDATA(**D***128)

10.3.6.2 Response if ProtMode = "0001_b": CBC encipherment only

Custom data is added with confidentiality protection.

The CBC mode (with AES in forward operation) is used to encipher all **D** custom data blocks.

KeyID identifies the encryption key for CBC encipherment with AES forward operation.

Ciphertext[1] is the encrypted challenge IChallenge_TAM2 (80 bits) of the Interrogator, after first prefixing the constant C_TAM2 and the randomly generated TRnd_TAM2: AES-ENC(Key[KeyID].ENC_key, C_TAM2[15:0] || TRnd_TAM2[31:0] || IChallenge_TAM2[79:0])

Table 8 — Response if ProtMode = "0001_b": CBC encipherment only

	TResponse
# of bits	(1+**D**)*128
Description	Ciphertext[1] || AES-CBC-ENC(IV = Ciphertext[1], Key[KeyID].ENC_key, CUSTOMDATA(**D***128))

10.3.6.3 Response if ProtMode = "0010_b": CMAC message authentication only

The CMAC mode is used to calculate a message integrity protecting code over the authentication cipher block and the **D** custom data blocks.

Parameter KeyID identifies key[KeyID].ENC_key for CBC encipherment and key[KeyID].MAC_key for CMAC computation.

Ciphertext[1] is the encrypted challenge IChallenge_TAM2 (80 bits) of the Interrogator, after first prefixing the constant C_TAM2 and the randomly generated TRnd_TAM2: AES-ENC(Key[KeyID].ENC_key, C_TAM2[15:0] || TRnd_TAM2[31:0] || IChallenge_TAM2[79:0])

AES-CMAC-96 is used to calculate the truncated 96-bit CMAC (with AES in forward operation) over the initial encrypted authentication block and the **D** following plaintext custom data blocks.

Table 9 — Response if ProtMode = "0010_b": CMAC message authentication only

	TResponse
# of bits	(1+**D**)*128+96
Description	Ciphertext[1] || CUSTOMDATA(**D***128) || AES-CMAC-96(Key[KeyID].MAC_key, Ciphertext[1] || CUSTOMDATA(**D***128))

10.3.6.4 Response if ProtMode = "0011_b": CBC encipherment with CMAC message authentication

Custom data is added with confidentiality and integrity protection.

The CBC mode (with AES in forward operation) is used to encipher the initial authentication block and all following **D** custom data blocks.

Parameter KeyID identifies key[KeyID].ENC_key for CBC encipherment and key[KeyID].MAC_key for CMAC computation.

Ciphertext[1] is the encrypted challenge IChallenge_TAM2 (80 bits) of the Interrogator, after first prefixing the constant C_TAM2 and the randomly generated TRnd_TAM2: AES-ENC(Key[KeyID].ENC_key, C_TAM2[15:0] || TRnd_TAM2[31:0] || IChallenge_TAM2[79:0])

Ciphertext[1+**D**] is the result of appending Ciphertext[1] with the encrypted custom data blocks: Ciphertext[1] || AES-CBC-ENC(IV= Ciphertext[1] Key[KeyID].ENC_key, CUSTOMDATA(**D***128))

AES-CMAC-96 is used to calculate (with AES in forward mode) the truncated 96-bit CMAC over the **D**+1 encrypted blocks.

Table 10 — Response if ProtMode = "0011_b": CBC encipherment with CMAC message authentication

	TResponse
# of bits	(1+**D**)*128+96
description	Ciphertext[1+**D**] || AES-CMAC-96(Key[KeyID].MAC_key, Ciphertext[1+**D**])

10.3.7 Final Interrogator processing TAM2

If ProtMode = 0010_b or ProtMode = 0011_b, the Interrogator first checks the supplied MAC for correctness and aborts if MAC verification fails. The Interrogator (or the external application controlling the Interrogator) then decrypts the TAM2 Response (TResponse) and shall verify whether: C_TAM2 and IChallenge_TAM2 have the correct value. If the values are correct, then the Tag can be considered as authentic. (The additional data for Profile[3:0]) can be accepted if C_TAM2 and IChallenge_TAM2 have the correct values.)"

11 Communication

This crypto suite does not support secure communication.

12 Key Table

This part of ISO/IEC 29167 defines a Key Table.

A Tag shall store one or more keys in the Key Table. A key is identified by the KeyID, the identification number of the key within the Key Table. KeyID shall start with "00_h" and increment with one for every next key in the Key Table.

Each key shall contain an encryption key (ENC_key).

Each key may contain a message authentication key (MAC_key).

Encryption keys shall be exclusively used for Tag authentication and encryption of additional data.

Message authentication keys shall be exclusively used for the authentication of additional data.

The Tag shall maintain a record in the Key Table for each key.

A record of the Key Management Table shall have the following fields for every key:

— KeyID: 8-bit identifier of the key in the Key Management Table

— RFU: 1-bit reserved for future use

— ENC_key: 128-bit Encryption Key that is used for Tag authentication and for the confidentiality protection (encryption/decryption) of additional custom data.

A record of the Key Management Table may have the following fields for every key:

— MPI: 16-bit Memory Profile Indicator.

Each key may be linked to a memory profile that is supported by the Tag. The links are stored in the MPI field. The MPI field contains 16 bits that correspond to a memory profile that is supported on the Tag and defines if a security command of the air interface has the right to use that key to authenticate the Tag, encrypt the custom data and/or authenticate the custom data for the specified memory profile. MPI[0:0] to MPI[15:15] refers to memory profile 0 to 15 respectively, as far as they are supported by the Tag. If the value of an MPI bit is "0_b" this key shall not be used by the related profile. If the value of an MPI bit is "1_b" this key may be used by the related profile.

Table 11 describes the link of each bit of MPI with a memory profile.

Table 11 — Link of MPI bits with memory profiles

MPI bit	Function
MPI[0:0]	Memory profile 00 has the right to use this Key
MPI[1:1]	Memory profile 01 has the right to use this Key
MPI[2:2]	Memory profile 02 has the right to use this Key
MPI[3:3]	Memory profile 03 has the right to use this Key
....	....
MPI[14:14]	Memory profile 14 has the right to use this Key
MPI[15:15]	Memory profile 15 has the right to use this Key

The MPI bit or bits for non-existing profile on a Tag shall be permalocked to zero (bit "0_b") by the Tag manufacturer.

MPI is an optional field, but is shall be supported if the Tag supports the TAM2 mode (with custom data)

— RFU: 1-bit reserved for future use

— MAC_key: 128-bit Message Authentication Key that may be used for the computation and validation of message authentication codes (MAC).

NOTE The ENC_key and MAC_key should be generated independently to support separation of different cryptographic functions.

Table 12 — Key Management Table

KeyID [a]	**RFU**	**ENC_key**	**MPI** (optional)	**RFU**	**MAC_key** (optional)
00_h	1	key[127:0]	MPI[15:0]	1	key[127:0]
01_h	1	key[127:0]	MPI[15:0]	1	key[127:0]
02_h	1	key[127:0]	MPI[15:0]	1	key[127:0]
....	....	....		....	....
nn_h	1	key[127:0]	MPI[15:0]	1	key[127:0]

[a] See the definition of KeyID in Clause 4.

The size and initial values of the Key Management Table and its mapping to their respective physical memory locations on the Tag shall be defined by the manufacturer.

Annex A
(normative)

Crypto Suite State transition tables

Table A.1 — Crypto Suite State transition table

Start State	Response	Action	Next State
Any	TAM1	Verify Tag key	**Initial**
Any	TAM2	Verify Tag key and add custom data	**Initial**

Any combination of Start States and Transitions not listed in Table A.1 shall result in an error and consequently a transition to the **Initial** state.

All other errors resulting from the execution of commands shall result in an error and consequently a transition to the **Initial** state.

Annex B
(normative)

Error conditions and error handling

A Tag that encounters an error during the execution of a crypto suite operation might send an error reply to the Interrogator. The details of these error replies are defined in the respective air interface standards.

This annex contains a listing of the Error Conditions that may result from the operation of this crypto suite. Annex E defines how to translate this error condition into an error code for the air interface.

Table B.1 — Error conditions

Crypto Suite Error Condition	Description
Cryptographic Error	Cryptographic error detected. This triggers a reset.
Memory Overrun	The command attempted to access a non-existent memory location.
Not Supported	The requested functionality is not supported by this crypto suite.
Other error	Miscellaneous error.

Annex C
(normative)

Cipher description

The Advanced Encryption Standard (AES) block cipher is described in detail in Reference [5].

Annex D
(informative)

Test vectors

D.1 References for AES test vectors

D.1.1 Test vectors for the AES algorithm

Test vectors for the AES algorithm can be found in:

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Additionally the original submission to NIST included test vectors as well:

http://csrc.nist.gov/archive/aes/rijndael/rijndael-vals.zip

D.1.2 online AES calculator

An online AES calculator can be found at: http://testprotect.com/appendix/AEScalc

Annex E
(normative)

Protocol specific information

E.1 General

E.1.1 Concept of exchanging Message and Response

For the implementation of this crypto suite an air interface protocol shall support security commands that allow the exchange of data between the Interrogator and the Tag that has this crypto suite implemented. The security command contains a message with parameters for the crypto suite. The reply of the Tag contains a response with the data that is returned by the crypto suite. An example of such data exchange for this crypto suite is depicted in Figure E.1.

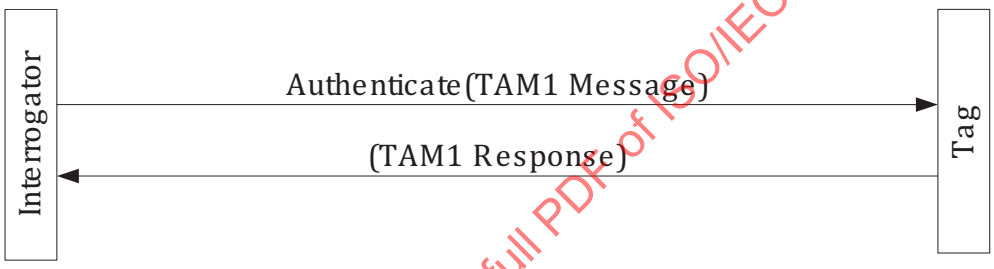

Figure E.1 — Message exchange for Tag authentication

The crypto suites that are defined by ISO/IEC 29167 can be defined by their Crypto Suite Identifier (CSI). According to ISO/IEC 29167-1 the CSI for this crypto suite shall be defined as the 6-bit value 000000_2. For use by the air interface protocols in this Annex the value is expanded to the 8-bit value 00_h.

This crypto suite is designed to provide security services for ISO/IEC 18000-3 mode 1, ISO/IEC 18000-3 mode 3 and ISO/IEC 18000-63. Details of the specific implementation for these air interface protocols are described in E.2, E.3 and E.4 respectively.

E.1.2 Supported security services

Table E.1 shows the security services that are supported by this crypto suite.

Table E.1 — Security services

Security Services	Method	Mandatory, optional, Prohibited, or not supported[a]
Authentication		Mandatory
Tag authentication (TA)		Mandatory
Interrogator authentication (IA)		Not supported
Mutual Authentication (MA)		Not supported
Communication		Not supported
Authenticated Tag from TA	Authenticated communication (Tag => Interrogator)	Not supported

[a] A crypto suite shall identify for each security service above and method if it is mandatory, optional, or prohibited

Table E.1 *(continued)*

Security Services	Method	Mandatory, optional, Prohibited, or not supported[a]
	Secure authenticated communication (Tag => Interrogator)	Not supported
Authenticated Interrogator from IA	Authenticated communication (Interrogator => Tag)	Not supported
	Secure authenticated communication (Interrogator => Tag)	Not supported

[a] A crypto suite shall identify for each security service above and method if it is mandatory, optional, or prohibited

E.2 Security services for ISO/IEC 18000-3 mode 1

Reserved for the implementation for ISO/IEC 18000-3 mode 1.

E.3 Security services for ISO/IEC 18000-3 mode 3

Reserved for the implementation for ISO/IEC 18000-3 mode 3.

E.4 Security services for ISO/IEC 18000-63

E.4.1 ISO/IEC 18000-63 protocol commands

A Crypto Suite supporting ISO/IEC 18000-63 shall fulfill the protocol security command requirements as defined in this section.

NOTE Optional choices shall be accepted for 1-to-1 communication. Reason: Since the Tag is singulated and the TID is known supported options can be derived from it.

1) The *Authenticate* command shall be supported.
2) The *Challenge* command shall not be supported.
3) The maximum execution time for an *Authenticate* Command containing a TAM1 or TAM2, payload shall be below 20 ms.
4) The Tag shall ignore commands from an Interrogator during execution of a cryptographic operation.
5) The Tag shall not support sending the contents of the ResponseBuffer in the reply to an ACK command.
6) The Tag shall support sending the contents of the ResponseBuffer in the reply to a *ReadBuffer* command
7) The Tag may support a security timeout following a crypto error.
8) A Tag in any cryptographic state other than initial (i.e. state after power-up) shall reset its cryptographic engine and transition to the open state upon receiving an invalid command. (Invalid commands means crypto commands with incorrect handle or CRC error)
9) For each Error Condition defined in the Crypto Suite:
   - The Tag shall transition to the **arbitrate** state
   - The Tag shall send an Error Code in case of a transition to the arbitrate state.
10) The Tag shall remain in its current state after a Tag Authentication.
11) This crypto suite does not support any encapsulation method.

E.4.2 Security commands in ISO/IEC 18000-63

In ISO/IEC 18000-63 the message to execute Tag authentication shall be transmitted to the Tag with the —*Authenticate*. The air interface shall return the response, either it shall be backscattered immediately after the command or it shall be stored in the ResponseBuffer, from where it shall be returned to the Interrogator with the *ReadBuffer* command.

NOTE Information about the *Authenticate* and *ReadBuffer* command and the ResponseBuffer can also be found in (reference to GS1 UHF EPC Gen2V2 document)

ISO/IEC 18000-63 specifies an 8-bit CSI. For implementation of this part of ISO/IEC 29167 in ISO/IEC 18000-63 the CSI shall be expanded to the 8-bit value 00_h.

E.4.3 Implementation of crypto suite error conditions in ISO/IEC 18000-63

This part of ISO/IEC 29167 specifies error conditions when the authentication is not successful. The error conditions of the crypto suite shall be returned to the Interrogator as error codes for the air interface. Table E.2 shows the conversion of Error Conditions in the crypto suite to ISO/IEC 18000-63 error codes.

Table E.2 — Implementation of crypto suite error conditions as Tag error codes

Crypto Suite Error condition	Description	ISO/IEC 18000-63 Error Code	ISO/IEC 18000-63 Error-Code Name
Cryptographic Error	Cryptographic error detected. This triggers a reset	00000101_b	Crypto Suite error
Memory Overrun	The command attempted to access a non-existent memory location	00000011_b	Memory overrun
Not Supported	The requested functionality is not supported by this crypto suite	00000001_b	Not supported
Other error	Miscellaneous error	00000000_b	Other error

E.4.4 Key properties

ISO/IEC 18000-63 requires the definition of key properties. If an implementation does provide key properties for a key belonging to this crypto suite it shall set the key properties to 0000_b.

E.4.5 Memory profiles

The table of memory profiles shall contain zero or more pointers to an area with custom data within the Tag's memory. The maximum number of pointers is 16.

Table E.3 — Description of ISO/IEC 18000-63 specific memory profiles for Profile

Value	Description
0000	globally defined memory profile: UII memory bank (starting at word position 0)
0001	globally defined memory profile: TID memory bank (starting at word position 0)
0010	globally defined memory profile: USER memory bank requested (file 0) (starting at word position 0)
0011	manufacturer-defined memory profile
....	....