INTERNATIONAL STANDARD

ISO/IEC 29167-16

Second edition
2022-11

Information technology — Automatic identification and data capture techniques —

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

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

Partie 16: Services de sécurité de la suite cryptographique ECDSA-ECDH pour les communications d'interfaces aériennes

Reference number
ISO/IEC 29167-16:2022(E)

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

Published in Switzerland

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.

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 or www.iec.ch/members_experts/refdocs).

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

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

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

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

This second edition cancels and replaces the first edition (ISO/IEC 29167-16:2015), which has been technically revised.

The main changes are as follows:

- certain normative references have been updated;
- editorial and technical revisions have been made to maintain conformance with ISO/IEC 18000-4:2018.

A list of all parts in the ISO/IEC 29167 series can be found on the ISO and IEC websites.

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

Introduction

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents or https://patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Automatic identification and data capture techniques —

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

1 Scope

This document describes a crypto suite based on elliptic curve cryptography (ECC) for the ISO/IEC 18000 series of standards protocol. In particular, this document specifies the use of elliptic curve Diffie-Hellman (ECDH) key agreement in a secure channel establishment and the use of elliptic curve digital signature algorithm (ECDSA) in an authentication mechanism.

This document specifies a crypto suite for ECDSA-ECDH for air interface for RFID systems. The crypto suite is defined in alignment with existing air interfaces.

This document defines a mutual authentication method 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. Key update is not supported in this document.

ECDSA-ECDH cipher is a high-weight security protocol especially for active RFID system, aiming at meeting those scenarios with high level security requirement.

2 Normative references

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

ISO/IEC 18000-4:2018, *Information technology — Radio frequency identification for item management — Part 4: Parameters for air interface communications at 2,45 GHz*

ISO/IEC 19762, *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*

ISO/IEC 14888-3, *IT Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms*

ISO/IEC 11770-3, *Information security — Key management — Part 3: Mechanisms using asymmetric techniques*

ISO/IEC 9797-3, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 3: Mechanisms using a universal hash-function*

ISO/IEC 9798-3, *IT Security techniques — Entity authentication — Part 3: Mechanisms using digital signature techniques*

ISO/IEC 18031, *Information technology — Security techniques — Random bit generation*

ISO/IEC 18033-3, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

ISO/IEC 11770-6, *Information technology — Security techniques — Key management — Part 6: Key derivation*

3 Terms and definitions

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

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

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

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

3.1
command
message that interrogator sends to tag with "Message" as parameter

3.2
Message
part of the *command* (3.1) that is defined by the crypto suite

3.3
reply
response that tag returns to the interrogator with "Response" as parameter

3.4
Response
part of the *reply* (stored or sent) (3.3) that is defined by the crypto suite

4 Symbols and abbreviated terms

4.1 Symbols

$xxxx_2$	Binary notation
$xxxx_h$	Hexadecimal notation
$\|\|$	Concatenation of syntax elements, transmitted in the order written
$()_{abscissa}$	Refers to that element of an ordered pair which is plotted on the horizontal axis of a two-dimensional cartesian coordinate system
$\bullet$	Point multiply

4.2 Abbreviated terms

CRC	Cyclic redundancy check
CS	Crypto suite
CSI	Cryptographic suite identifier
DSA	Digital signature algorithm

EBV	Extensible bit vector
ECC	Elliptic curve cryptography
ECDH	Elliptic curve Diffie-Hellman
ECDHP	ECDH parameter
ECDSA	Elliptic curve digital signature algorithm
FN	Fragmentation number
IAK	Integrity authentication key
IID	IDentifier of interrogator
MIC	Message integrity check code
MAC	Message authentication code
MAM	Mutual authenticate message
MK	Master key
MTU	Maximum transmission unit
RFU	Reserved for future use
RN	Random number
RFID	Radio frequency identification
SEK	Session encryption key
SIK	Session integrity check key
TID	IDentifier of tag
TPK	Temporary public key
TRAIS	Tag and reader air interface security
TRAIS-P	Tag and reader air interface security based on public key cryptography
TTP	Trusted third party
TTPID	IDentifier of TTP

5 Conformance

5.1 Claiming conformance

To claim conformance with this document, an Interrogator or a Tag shall comply with all relevant clauses of this document except those marked as “optional”.

Relevant conformance test methods are provided in ISO/IEC 19823-16[1].

5.2 Interrogator conformance and obligations

To conform to this document, an Interrogator shall implement the mandatory messages and responses format defined in this document, and conform to the relevant part of the ISO/IEC 18000 series.

To conform to this document, an Interrogator may implement any subset of the optional parameters for message and response format defined in this document.

To conform to this document, the Interrogator shall not

- implement any messages and responses format that conflicts with this document, or
- require the use of an optional, proprietary, or custom parameters for message and response format to meet the requirements of this document.

5.3 Tag conformance and obligations

To conform to this document, a Tag shall implement the mandatory message and response formatting defined in this document for the supported types, and conform to the relevant part of the ISO/IEC 18000 series.

To conform to this document, a Tag may implement any subset of the optional parameters in the message and response formatting defined in this document.

To conform to this document, a Tag shall not

- implement any message and response formatting that conflicts with this document, or
- require the use of an optional, proprietary, or custom parameter in the message and response formatting to meet the requirements of this document.

6 Cipher introduction

The ECDSA is a variant of the DSA which uses ECC. ECDSA supports mutual authentication and has been specified in ISO/IEC 14888-3. The cipher descriptions of Annex C shall apply.

ECDH is an anonymous key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. This shared secret shall be directly used as a key, or better yet, to derive another key which shall then be used to encrypt subsequent communications using a symmetric key cipher. It is a variant of the Diffie-Hellman protocol using ECC. ECDH protocol specified in ISO/IEC 11770-3 shall apply.

ECC is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Compared to the RSA algorithm, ECC offers equivalent security with smaller key sizes which result in savings for power, memory, bandwidth, and computational resources that make ECC especially attractive for RFID system.

7 Parameter definitions

7.1 Parameter definitions

Table 1 contains the parameters definitions of the crypto suite.

Table 1 — Descriptions of parameters

Parameter	Description
FN[7:0]	The number of fragmentations.
AuthType[1:0]	This shows the authentication type in the authentication procedure. The values are as following: — 00: mutual authentication; — 01: reserved for the use of interrogator authentication; — 10: reserved for the tag authentication; — 11: Other (as defined by the CSI).

AuthStep[2:0]	This shows the step number in the authentication procedure. The values are as following:
	— 000: Step 1 of Authenticate command;
	— 001: Step 2 of Authenticate command;
	— 010-111: All other values are RFU.

ECDHP[255:0]	ECDH parameter, consist of parameter ID, parameter length and parameter content three parts, where the parameter ID shall be 8 bits; parameter shall be 16 bits in length and indicates the number of bytes in the parameter content. The values of ECDH parameter: 1) 01h: The field value shall be denoted by OIDs. The Length subfield indicates the number of octets of OIDs. The values of Content subfield are the content of OIDs. 2) Other: All other values are RFU.
Certx[Variable]	The digital certificate of x. x can be tag, interrogator or TTP. See 7.2.
RNt[63:0]	64-bit random number generated by the tag.
X t[391:0]	Temporary private key generated by tag and used for ECDH exchange.
TPKt[391:0]	Temporary public key generated by tag and used for ECDH exchange, the procedure of generation is as follows: the tag generates a temporary private key which is used for ECDH exchange, and temporary public key TPKt=Xt•P.
TTPID[Variable]	Specifying whether or not the TTP is to be involved and the identifier of the TTP.
Sigt[383:0]	Digital signature generated by the tag.
RNi[63:0]	64-bit random number generated by the interrogator.
Xi[391:0]	Temporary private key generated by interrogator and used for ECDH exchange.
TPKi[391:0]	Temporary public key generated by interrogator and used for ECDH exchange, the procedure of generation is as follows: the interrogator generates a temporary private key which is used for ECDH exchange, the temporary public key TPKi=Xi•P.
MICi[255:0]	Message integrity code generated by the interrogator.
Sigi[383:0]	Digital signature generated by the interrogator.
MICt[255:0]	Message integrity code generated by the tag.
MK[127:0]	Master key.
AuthRes[Variable]	Authentication result generated by the TTP and contains the value of RESt, RESi and Sigttp.

ECC parameters example see Annex G.

7.2 Certiticate format

Figure 1 specifies the encoding of digital certificate Cert$_x$ in the TLV format.

	Cert Type	Cert Length	Value
# of bits	4	12	variable

Figure 1 — Certificate format

The Cert Type subfield specifies the type of the certificate and shall be 4 bits in length. The values are:

a) 0000: Value subfield contains X.509 certificate of Interrogator, Cert_i;

b) 0001: Value subfield contains X.509 certificate of Tag, Cert_t;

c) 0010: Value subfield contains X.509 certificate of TTP, Cert_{ttp};

d) Other: All other values are RFU.

The 12-bit Cert Length subfield contains the length in number of octets of the Value subfield, in the range of 1 to 4095.

8 State diagram

The state diagram for this cryptographic suite consists of four states. The transition between these states is specified in Figure 2. The state transition table of Annex A shall apply.

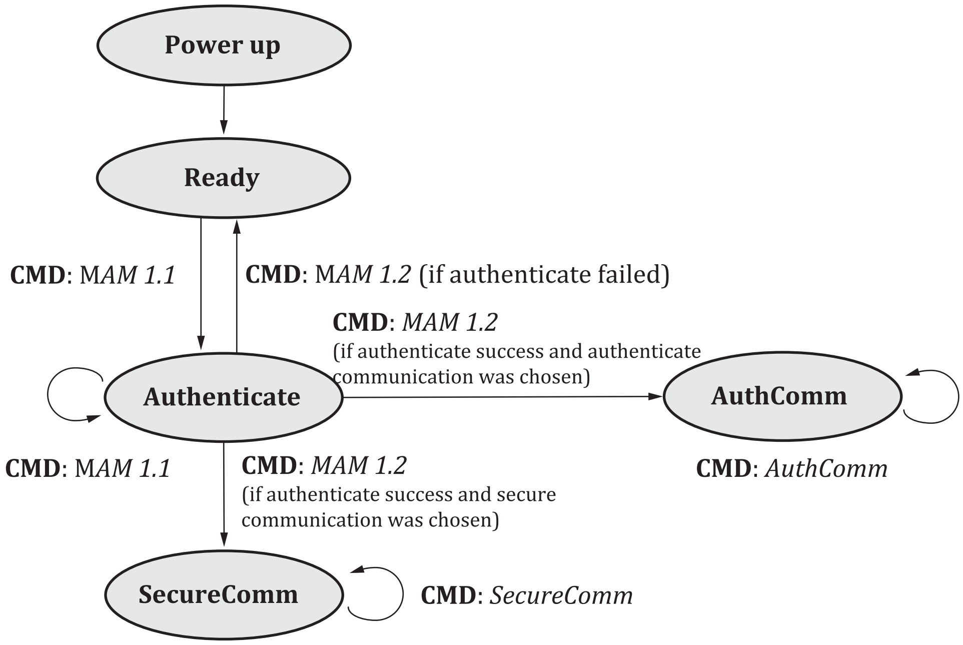
Power up
Ready
CMD: MAM 1.1
CMD: MAM 1.2 (if authenticate failed)
CMD: MAM 1.2
(if authenticate success and authenticate communication was chosen)
Authenticate
AuthComm
CMD: MAM 1.1
CMD: MAM 1.2
(if authenticate success and secure communication was chosen)
CMD: AuthComm
SecureComm
CMD: SecureComm

Figure 2 — State diagram

9 Initialization and resetting

This document shall implement Ready, Authenticate, AuthComm and SecureComm states.

After power-up and after a reset of the crypto suite the tag moves into the Ready state.

Implementations of this 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 General

This document describes additions to the ISO/IEC 18000 series of standards protocol to support the TRAIS-P. Especially, it defines

a) the use of ECC certificates and ECDSA for mutual authentication of an interrogator and a tag,

b) the use of the ECDH key agreement scheme with keys to establish the secure channel, and

c) the encoding in the related commands and the processing of those messages.

Figure 3 shows protocol flows of ECDSA-based mutual authentication procedure with the key agreement of ECDH.

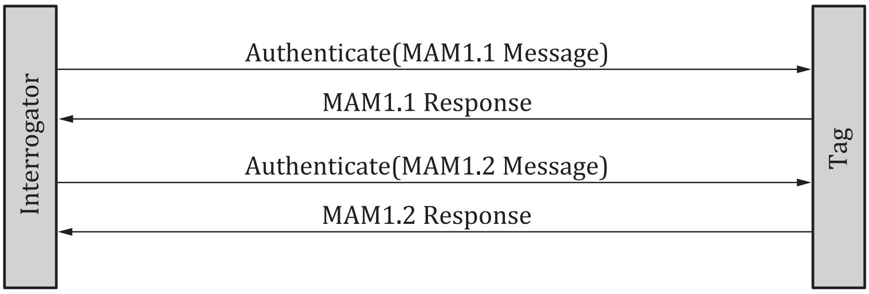

Figure 3 — Mutual authentication with key agreement

The mechanism is based on the ISO/IEC 9798-3, the interrogator and tag can also involve an on-line trusted third party for the mutual authentication, Figure 4 shows protocol flows between on-line trusted third party and an interrogator (for the case of TTP involving, Annex H shall apply).

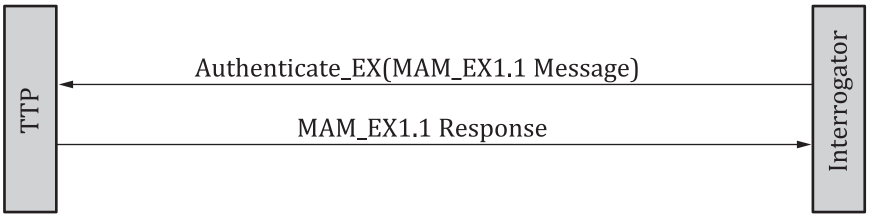

Figure 4 — Protocol flows between TTP and interrogator

10.2 Authenticate message

10.2.1 Message in Authenticate command and reply

Interrogators and Tags shall implement the Authenticate command, message in Authenticate command as shown in Table 2. The fast response in reply to an Authenticate command is shown in Table 3. An Interrogator uses Authenticate commands to perform mutual authentication. The CSI specified in the message selects a particular cryptographic suite from among those supported by the Tag.

Table 2 — Message in Authenticate command

	CSI	Length	Message
# of bits	8	EBV	Variable
description	CSI	length of message	message (depends on CSI)

Table 3 — Fast response in reply to an Authenticate command

	Length	Response
# of bits	EBV	Variable
description	length of response	response (depends on CSI)

10.2.2 Authenticate(MAM1.1 Message)

The message of Authenticate command of MAM1.1 is as shown in Table 4.

Table 4 — MAM1.1 Message

	Message						
	FN	IID	Auth Type	Auth Step	TTPID	Cert_i	ECDHP
# of bits	8	64	2	3	Variable	Variable	256
description	fragmentation number	interrogator identifier	00	000	TTP involved or not	digital certificate of interrogator	ECDH parameter

The fields of MAM1.1 Message shall have the following meaning:

a) FN: This field shall be 8 bits in length and specifies the number of fragmentations (protocol specific of Annex E shall apply).

b) IID: This field shall be 64 bits in length and specifies the Interrogator identifier.

c) AuthType: This field shall be 2 bits in length and the values of the AuthType field are as follows:

- 00: mutual authentication;
- 01: reserved for the use of interrogator authentication;
- 10: reserved for the tag authentication;
- 11: RFU.

d) AuthStep: This field shall be 3 bits in length and specifies the step number in the procedure. Each authentication procedure requires a pre-determined number of steps. In MAM1.1 Message, the value is 000.

e) TTPID: Bit [7:0] of this field specifies whether or not the TTP shall be involved by the interrogator in the mutual authentication. The optional bit [71:8] is only present and shall be the identifier value of the TTP while bit [7:0] is set to 0000 0001 (for the case of TTP involving, Annex H shall apply). The values of bit [7:0] of the TTP field are as follows:

- 0000 0000: TTP not to be involved;
- 0000 0001: TTP to be involved;
- other: All other values are RFU.

f) Cert_i: This field specifies the digital certificate of interrogator. See 7.2.

g) ECDHP: This field shall be 256 bits in length and specifies the ECDH parameter, consisting of parameter ID, parameter length and parameter content. Where the parameter ID shall be 8 bits; parameter length shall be 16 bits in length and indicates the number of bytes in the parameter content. The values of ECDH Parameter:

1) 01h: The field value shall be denoted by OIDs. The Length subfield indicates the number of octets of OIDs. The values of Content subfield are the content of OIDs.

2) Other: All other values are RFU.

10.2.3 MAM1.1 Response

The response of MAM1.1 is as shown in Table 5.

Table 5 — MAM1.1 Response format

	Response							
	FN	**TID**	**TTPID**	**Cert_t**	**RN_t**	**TPK_t**	**ECDHP**	**Sig_t**
# of bits	8	64	Variable	Variable	64	392	256	384
description	fragmentation number	tag identifier	TTP involved or not	digital certificate of tag	random number of tag	temporary public key for ECDH	ECDH parameter	ECDSA signature of tag

The fields of MAM1.1 Response shall have the following meaning:

a) FN: This field shall be 8 bits in length and specifies the number of fragmentations (the protocol specific of Annex E shall apply).

b) TID: This field shall be 64 bits in length and specifies the Tag identifier.

c) TTPID: This field specifies whether or not the TTP is to be involved by the interrogator in the mutual authentication. The value is the same as the one in the MAM1.1 Message. For the case of TTP involving, Annex H shall apply.

d) Cert_t: This filed specifies the digital certificate of tag. See 7.2.

e) RN_t: This field shall be 64 bits in length and specifies the random number generated by the tag. The random number generation method specified in ISO/IEC 18031 shall apply.

f) TPK_t: This field shall be 392 bits in length and specifies the temporary public key generated by the tag and used for ECDH exchange. A tag generates a temporary private key X_t which is used for ECDH exchange, then computes the temporary public key $\text{TPK}_t = X_t \bullet P$.

g) ECDHP: This field shall be 256 bits in length and specifies the ECDH parameter, consisting of parameter ID, parameter length and parameter content. Where the parameter ID shall be 8 bits; parameter length shall be 16 bits in length and indicates the number of bytes in the parameter content. The values of ECDH Parameter:

  1) 01_h: the field value shall be denoted by OIDs. The Length subfield indicates the number of octets of OIDs. The values of Content subfield are the content of OIDs;

  2) Other: all other values are RFU.

h) Sig_t: This field shall be 384 bits in length and specifies the digital signature generated by the tag. The value is computed by $\text{Sig}_t = \text{ECDSA}(S_t, \text{TID}||\text{IID}||\text{Cert}_t||\text{TTPID}||\text{RN}_t||\text{TPK}_t||\text{ECDHP})$.

NOTE This document only describes ECC-192. An update of the length of command parameters are required when use of other ECC curves.

10.2.4 Authenticate(MAM1.2 Message)

The message of Authenticate command of MAM1.2 is as shown in Table 6.

Table 6 — MAM1.2 Message

	Message								
	FN	Auth Type	Auth Step	RN_t	RN_i	TPK_i	Sig_i	MIC_i	AuthRes
# of bits	8	2	3	64	64	392	384	256	Variable
description	fragme-nta-tion number	00	001	random number of tag	random number of inter-ro-gator	temporary public key for ECDH	Interro-gator ECDSA signa-ture	message integrity code	results of TTP authen-ti-cation

The fields of MAM1.2 Message shall have the following meaning:

a) FN: This field shall be 8 bits in length and specifies the number of fragmentations (the protocol specific of Annex E shall apply).

b) AuthType: This field shall be 2 bits in length and the value of the AuthType field shall be the same as the one in the MAM1.1 Message. The descriptions of the values are:

- 00: mutual authentication;
- 01: reserved for the use of interrogator authentication;
- 10: reserved for the tag authentication;
- 11: other (as defined by the CSI).

c) AuthStep: This field shall be 3 bits in length and specifies the step number in the procedure. Each authentication procedure requires a pre-determined number of steps. In MAM1.2 Message, the value is 001.

d) RN_t: This field shall be 64 bits in length and specifies the random number generated by the tag. The value is the same as the one in the MAM1.1 Response. The random number generation method specified in ISO/IEC 18031 shall apply.

e) RN_i: This field shall be 64 bits in length and specifies the random number generated by the interrogator.

f) TPK_i: This field shall be 392 bits in length and specifies the temporary public key generated by the interrogator and used for ECDH exchange. An interrogator generates a temporary private key X_i which is used for ECDH exchange, then computes the temporary public key $TPK_i=X_i \bullet P$.

g) Sig_i: This field shall be 384 bits in length and specifies the digital signature generated by the interrogator. The value is computed by $Sig_i=ECDSA(S_i,TID||IID||TTPID||RN_t||RN_i||TPK_i)$.

h) MIC_i: This field shall be 256 bits in length and specifies the message integrity code generated by the interrogator. The value is computed by $MIC_i=HMAC\text{-}SHA256(IAK, TID||IID||TTPID||RN_t||RN_i||TPK_i||Sig_i)$.

i) AuthRes: This field is optional and shall be present while bit [7:0] of the TTPID field is 0000 0001, otherwise, this field is not present. For the case of TTP involving, Annex H shall apply.

10.2.5 MAM1.2 Response

The response of MAM1.2 is as shown in Table 7.

Table 7 — MAM1.2 Response format

	Response		
	FN	**RN$_i$**	**MIC$_t$**
# of bits	8	64	256
description	fragmentation number	random number of interrogator	message Integrity code

The fields of MAM1.2 Response shall have the following meaning:

a) FN: This field shall be 8 bits in length and specifies the number of fragmentations (the protocol specific of Annex E shall apply).

b) RN$_i$: This field shall be 64 bits in length and specifies the random number generated by the interrogator. The value is the same as the one in the MAM1.2 Message.

c) MIC$_t$: This field shall be 256 bits in length and specifies the message integrity code generated by the Tag. The value is computed by MIC$_t$=SHA256(IAK, TID||IID||RN$_i$).

10.3 Authentication procedure

10.3.1 Protocol requirements

Based on the ISO/IEC 9798-3, the authentication protocol requires the tag should have ECC-based private key S$_t$ and the related certificate Cert$_t$. The interrogator shall have ECC-based private key S$_i$ and the related certificate Cert$_i$.

10.3.2 Procedure

Authentication protocol flows of ECDSA-based mutual authentication procedure with the key agreement of ECDH are as follows:

a) The interrogator transmits Authenticate(MAM1.1 Message) to the tag. Where, the TTPID, Cert$_i$ and ECDHP are included.

b) After receiving Authenticate (MAM1.1 Message), the tag confirms whether or not to involve the TTP based on the value of TTPID field, if the TTP policy does not match, ignore the message and the authentication procedure is failed. Otherwise, transmit MAM1.1 Response to the interrogator, including the TTPID, certificate Cert$_i$, random number RN$_t$ generated by the tag, temporary public key TPK$_t$ and ECDHP used for ECDH exchange and generated by the tag, and digital signature Sig$_t$ generated by the tag by using its own private key to compute TID||IID||Cert$_t$||TTPID||RN$_t$||TPK$_t$||ECDHP, Sig$_t$=ECDSA (S$_t$,TID||IID||Cert$_t$||TTPID||RN$_t$||TPK$_t$||ECDHP).

c) After the interrogator has received MAM1.1 Response, the operation is as follows:

   1) Confirm whether the values of TTPID and ECDHP in MAM1.1 Response are equal to the values of TTPID and ECDHP in Authenticate(MAM1.1 Message), respectively, if not, ignore the response, the authentication procedure is failed. Otherwise, use the tag's public key Q$_t$ extracted from certificate Cert$_t$ to verify the tag's signature Sig$_t$. If the signature verification is failed, ignore the response, the authentication procedure is failed. The error code type of Annex B shall apply. Otherwise, go to step 2).

   2) The interrogator generates temporary private key X$_i$ and temporary public key TPK$_i$, where TPK$_i$=X$_i$•P. and uses X$_i$ and TPK$_t$ to perform the ECDH computation, and gets the primary key seed (X$_i$•TPK$_t$) $_{\text{abscissa}}$, where (X$_i$•TPK$_t$)$_{\text{abscissa}}$ indicates X coordinate of X$_i$•TPK$_t$, and X$_i$•TPK$_t$ shall not be infinite point. The interrogator computes KD-HMAC-SHA256((X$_i$•TPK$_t$) $_{\text{abscissa}}$,RN$_t$||RN$_i$) to generate a random value that is used as Master Key (MK), 128 bits. Go to step 3). The KD-HMAC-SHA256 function specified in ISO/IEC 11770-6 shall apply.

3) The interrogator computes KD-HMAC-SHA256 (MK, TID||IID||RN_t||RN_i) to generate a random value, where the first part is used as integrity authentication key IAK, 128 bits, the second part is used as session integrity check key SIK, 128 bits, the third part is used as session encryption key SEK,128 bits.

d) The interrogator transmits Authenticate (MAM1.2 Message) to the tag. The command includes the random number RN_t generated by the tag, random number RN_i generated by the interrogator, temporary public key used for ECDH and generated by the interrogator, digital signature Sig_i generated by the interrogator to use its own private key to compute TID||IID||TTPID||RN_t||RN_i||TPK_i, Sig_i=ECDSA(S_i,TID||IID||TTPID||RN_t||RN_i||TPK_i), and message integrity check code (MIC) generated by the interrogator to use its IAK to compute TID||IID ||TTPID||RN_t||RN_i||TPK_i||Sig_i, MIC_i=HMAC-SHA256(IAK,TID||IID||TTPID||RN_t||RN_i||TPK_i||Sig_i).

e) After receiving Authenticate (MAM1.2 Message), the tag should operate as follows:

1) Confirm whether the value of random number RN_t in Authenticate (MAM1.2 Message) is equal to the value of random number RN_t in MAM1.1 Response, if not, ignore the command, the authentication procedure is failed; Otherwise, use the interrogator's public key Q_i extracted from the certificate $Cert_i$ to verify the digital signature Sig_i. If the signature verification is failed, ignore the response, the authentication procedure is failed. Otherwise, go to step 2).

2) The tag performs the ECDH computation on X_t and TPK_r, to get the primary key seed $(X_t \bullet TPK_i)_{abscissa}$, where $(X_t \bullet TPK_i)_{abscissa}$ indicates X coordinate of $X_t \bullet TPK_i$, and $X_t \bullet TPK_i$ shall not be infinite point. The tag computes KD-HMAC-SHA256($(X_t \bullet TPK_i)_{abscissa}$,$RN_t$||$RN_i$) to generate a random value that is used as the MK, 128 bits. Go to step 3).

3) The tag computes KD-HMAC-SHA256(MK, TID||IID||RN_t||RN_i) to generate a random value, where the first part is used as an integrity authentication key IAK, the second part is used as session integrity check key SIK, the third part is used as a session encryption key SEK. Go to step 4).

4) The tag computes TID||IID||TTPID||RN_t||RN_i||TPK_i||Sig_i to get the message integrity check code MIC using its IAK, MIC_i=HMAC-SHA256(IAK, TID||IID||TTPID||RN_t||RN_i||TPK_i||Sig_i), and bit-wise compares the received MIC in the Authenticate command against the computed MIC. If they differ in any bit position, then ignore the command; the authentication procedure is failed. Otherwise, the tag authentication process to the interrogator is successful. Go to step 5).

5) The tag transmits MAM1.2 Response to the interrogator, including the random number RN_i generated by the interrogator, and message integrity check code MIC generated by the tag to use its IAK to compute TID||IID||RN_i, MIC_t=HMAC-SHA256(IAK, TID||IID||RN_i).

f) After receiving MAM1.2 Response, the interrogator first confirms whether the value of random number RN_i in the response is equal to the value of random number RN_i in the Authenticate (MAM1.2 Message), if not, ignores the response, the authentication procedure is failed. Otherwise, computes TID||IID||RN_i to get the MIC using its IAK, MIC_t=HMAC-SHA256(IAK, TID||IID||RN_i), and bit-wise compares the received MIC in the response against the computed MIC, if they differ in any bit position, then ignore the response; the authentication procedure is failed. Otherwise, the authentication process is successful.

Annex D provides an example of test vectors of the authentication protocol.

When the message shall partition into smaller fragments, the methods of message's fragmentation and defragmentation of Annex F shall apply.

The certificate status verification shall be performed by both the tag and interrogator after they received the certificate from each other and include verification of the certificate's authenticity and expiration status. See RFC 3280[2] for more information. The validation of certificate revocation status is optional and may be performed by checking against a certificate revocation list or by contacting an OCSP responder.

In order to resist some security attack such as man-in-the-middle attack, the FN, AuthType, AuthStep, IID and TID in the received messages also shall be checked by both the tag and interrogator.

11 Communication

11.1 Authenticate communication

Figure 5 shows a representative procedure for an interrogator sending or receiving data using authenticate communications.

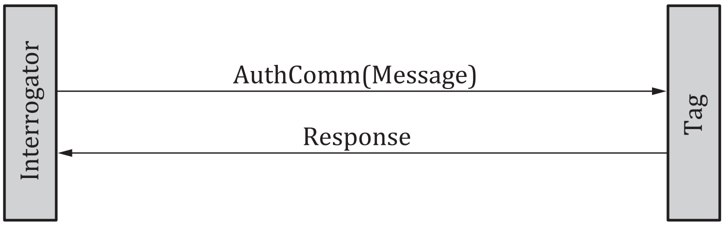

Figure 5 — Authenticate communication

After a tag has cryptographically authenticated an interrogator, it may accept subsequent commands encapsulated in the AuthComm command. The message in an AuthComm shown in Table 8, it includes CSI field, Length field, Message field and MAC field. The Message field shall encapsulate a tag-supported command. Before encapsulating a command, an Interrogator shall remove the preamble, handle and CRC from the command. An interrogator includes a MAC in the AuthComm, the MAC is the message authentication code of the message in AuthComm. The MAC mechanism specified in ISO/IEC 9797-3 shall apply. The response in reply to an AuthComm command is shown in Table 9, the MAC in the response is the message authentication code of Length and Response. The error code type of Annex B shall apply.

Table 8 — Message in AuthComm command

	CSI	Length	Message	MAC
# of bits	8	EBV	Variable	128
description	CSI	length of Message	message	message authentication code

Table 9 — Response in reply to an AuthComm command

	Length	Response	MAC
# of bits	EBV	Variable	128
description	length of Response	response	message authentication code

11.2 Secure communication

Figure 6 shows a representative procedure for an interrogator sending or receiving data using secure communications. For secure communication, the symmetric encryption algorithm specified in ISO/IEC 18033-3 shall apply.

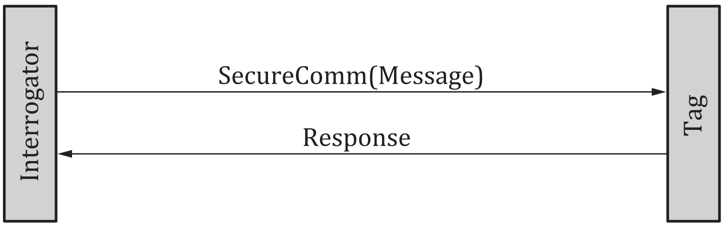

Figure 6 — Secure communication

After a tag has cryptographically authenticated an interrogator, it shall accept subsequent commands encapsulated in the SecureComm command. The message in a SecureComm shown in Table 10, it includes CSI field, Length field, Message field and an optional MAC field. The Message field shall encapsulate an encrypted tag-supported command. Before encapsulating a command an Interrogator shall remove the preamble, handle, and CRC from the command. An interrogator encrypts the message and/or shall include a MAC in the SecureComm, the MAC is the message authentication code of response in SecureComm. The response in reply to a SecureComm command is shown in Table 11. The error code type of Annex B shall apply. The MAC in SecureComm and the related response commands is optional and shall be present while Bit [7:0] of the TTPID field is 0000 0001.

Table 10 — Message in SecureComm command

	CSI	Length	Message	MAC
# of bits	8	EBV	Variable	128 (optional)
description	CSI	length of Message	Encrypted command (depends on CSI)	message authentication code

Table 11 — Response in reply to a SecureComm command

	Length	Response	MAC
# of bits	EBV	Variable	128 (optional)
description	length of response	encrypted response	message authentication code

Annex A
(normative)

State transition table

A.1 Ready state transition table

Ready state transition table shall be as shown in Table A.1.

Table A.1 — Ready state transition table

Command	Conditions	Next state
MAM1.1	All	Authenticate

A.2 Authenticate state transition table

Authenticate state transition table shall be as shown in Table A.2.

Table A.2 — Authenticate state transition table

Command	Conditions	Next state
MAM1.1	All	Authenticate
MAM1.2	Fail	Ready
MAM1.2	Success and authenticate communication was chosen	AuthComm
MAM1.2	Success and secure communication was chosen	SecureComm

A.3 AuthComm state transition table

AuthComm state transition table shall be as shown in Table A.3.

Table A.3 — AuthComm state transition table

Command	Conditions	Next state
AuthComm	All	AuthComm

A.4 SecureComm state transition table

SecureComm state transition table shall be as shown in Table A.4.

Table A.4 — SecureComm state transition table

Command	Conditions	Next state
SecureComm	All	SecureComm

Annex B
(normative)

Error codes and error handling

B.1 Error code format

Error code format shall be as shown in Table B.1.

Table B.1 — Error code format

Error type	Error subcode
8-bit	8-bit

B.2 Error type and error subcode

Error type and error subcode shall be as shown in Table B.2.

Table B.2 — Error type and error subcode

Error type	Description	Error sub-code	Description
01_h	Authentication failed	01_h	The authentication was failed.
02_h	Secure communication failed	01_h	The secure communication between interrogator and tag was failed.
03_h	Authenticate communication failed	01_h	The authenticate communication between interrogator and tag was failed.

Annex C
(normative)

Cipher description

The ECDSA is a variant of the DSA which uses ECC. The algorithm specified in ISO/IEC 14888-3 shall apply. Its security is based on the computational intractability of the discrete logarithm problem.

ECDH is an anonymous key agreement protocol that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel. This shared secret shall be directly used as a key, or better yet, to derive another key which can then be used to encrypt subsequent communications using a symmetric key cipher. ECDH has been specified in ISO/IEC 11770-3.

For the ECDH algorithm used in this document, there are some explanations as follows.

a) Temporary private keys x and y are the integers between 1 and n-1, where n is the degree of the base point P in the elliptic curve domain parameters.

b) Temporary pubic keys $x·P$ and $y·P$ are the points in the elliptic curve defined in the elliptic curve domain parameters.

c) The key seed $(x·y·P)_{\text{abscissa}}$ negotiated by ECDH is the x-coordinate at $x·y·P$. $x·y·P$ cannot be an infinite point.

Annex D
(informative)

Test vectors

D.1 Authentication elliptic *E* curve

The elliptic *E* curve for this example is curve P-192 as below:

E: $Y^2 = X^3 + \mathrm{a}X + b$

p = BDB6F4FE3E8B1D9E0DA8C0D46F4C318CEFE4AFE3B6B8551F

a = BB8E5E8FBC115E139FE6A814FE48AAA6F0ADA1AA5DF91985

b = 1854BEBDC31B21B7AEFC80AB0ECD10D5B1B3308E6DBF11C1

Base point *G* over *E*.

$G = (x_G, y_G)$

P=(4AD5F7048DE709AD51236DE65E4D4B482C836DC6E4106640,

02BB3A02D4AAADACAE24817A4CA3A1B014B5270432DB27D2)

n is the order of point *P*.

n = BDB6F4FE3E8B1D9E0DA8C0D40FC962195DFAE76F56564677

The bit length of the crypto private key is $\sigma = |n| = 192$ bits.

D.2 Authentication parameters

The parameters are as following:

Tag Private key

PrikeyTag = 96EB1F1FC18386012D20D630B613B6196CBAEE26687B3CD4

Tag Public Key point

$PubkeyTag = (04, x_{Tag}, y_{Tag})$

= (04,

A2FA97A41B88FB9552D5CA8EEA325EE51EB4ADAF88452801,

3CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878)

Tag Temporary Private Key

TempPrikeyTag = 386B55CAD70F1259484A688DDCC77A685DFCE36D227BDCB5

Tag Temporary Public point

$TempPubkeyTag = (04, x_{Tag}, y_{Tag})$

= (04,

`8BB733FA69A4EE35D37C9C7205615B0A767BD0B156B6811A,`

`8EE73F2EC633F8107245F00960526FEEB10273E5BA5AD290)`

Interrogator Private key

PrikeyInterrogator = `5997F9D758E0D2185F91B3EFA9ABC0668C9C617C2E85DBF0`

Interrogator Public Key point

PubkeyInterrogator = $(04, x_{Interrogator}, y_{Interrogator})$

= (04,

`BBA79921268948117DA7345B44479A3EBDF7FE54EE26EBF1,`

`A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A41429C)`

Interrogator Temporary Private Key

TempPrikeyInterrogator = `8B60CEF3ECBD2AB0F2E00F03C27176767D77B99BB451C47F`

Interrogator Temporary Public point

TempPubkeyInterrogator = $(04, x_{Interrogator}, y_{Interrogator})$

= (04,

`36399F5558CF34A6E519DA53681525216216165CFA37A903,`

`87068C4B9F8E1340339F507E999B9F783061A581BBED1116)`

TTP Private key

PrikeyTTP = `AD4CF6DD36FC4EB7F9A4D114DB1E81584E5551839C066FC5`

TTP Public Key point

PubkeyTTP = $(04, x_{TTP}, y_{TTP})$

= (04,

`9976B163C7BE616FFB84CF05FEB570C2B91D5270D298FD83,`

`012CCF376A217003BA25C6EC6F223E98CEAA53CDB7851F6B)`

D.3 Authentication process

Authentication process are as follows, where IID is `1364615367812479`, TID is `117A57A1735BB9C6` and TTPID is `0031143012060A0992`.

The interrogator sends MAM1.1 Message to the tag and receives the response from the tag. The MAM1.1 Message and Response are as follows.

[MAM1.1 Message]

```
FN(1octet)=01

IID(8octets)= 1364615367812479

AuthType(2bits)=00

AuthStep(3bits)=000

TTPID(9octets)=0031143012060A0992
```

Certi(411octets)=10993082019530820149A00302010202046390D2F5300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A130430303033310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D3134313030313136333333325A1 70D3135313031343136333334335A305931143012060A0992268993F
22C6401191604574 15049310B3009060355040613024 34E310D300B060355040A130430303033310B3009060
355040B1302534E31183016060355040314 0F696E746572726F6761746F72404145304A3
01406072A8648CE3D020106092A811CD763010102010332000 4BBA79921268948117DA73
45B44479A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A414
29CA314301230100609608648 0186F842010D0403160178300C06082A811CD7630101010
5000338003035021900 8FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F02183
1B27EA8752C2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA

ECDHP(32octets)=01000B06092a811cd7630101020100000000000000000000000000000 00000000

[MAM1.1 Response]

FN(1octet)=01

TID(8octets)= 117A57A1735BB9C6

TTPID(9octets)=0031143012060A0992

Cert$_t$(403octets)=11913082018D30820142A00302010202044A6C47C3300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A130430303033310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D3134313030313136333333325A1 70D3135313031343136333334335A305231143012060A0992268993F
22C6401191604574 15049310B3009060355040613024 34E310D300B060355040A130430303033310B3009060
355040B1302534E3111300F060355040314087461674041535545304A301406072A86
48C E3D020106092A811CD763010102010332000 4A2FA97A41B88FB9552D5CA8EEA325EE51EB
4ADAF884528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A3143012301
00609608648 0186F842010D0403160178300C06082A811CD763010101050003370030340
218259C42F134F22B96399F13C7C3106A0C103966B0F326384502180EEAC454472711285
6A44F7177D6D1E1BB3A9FEFEFB6C00C

RN$_t$(8octets)=53CC8496DF6878A4

TPK$_t$(49octets)=048BB733FA69A4EE35D37C9C7205615B0A767BD0B156B6811A8EE73F2
EC633F8107245F00960526FEEB10273E5BA5AD290

ECDHP(32octets)=01000B06092a811cd7630101020100000000000000000000000000000 00000000

Sig$_t$(48octets)=B0B294ECC5F1DB9C85B5EBDCF89AA47F359F4526DC33843628786AE35
E94DA884C2CF430C2C8D813CE5FF40E949241FA

The interrogator sends MAM1.2 Message to the tag and receives the response from the tag. The MAM1.2 Message and Response are as following:

[MAM1.2 Message]

FN(1octet)=01

AuthType(2bits)=00

AuthStep(3bits)=001

RN$_t$(8octets)=53CC8496DF6878A4

RN$_i$(8octets)=8BD3AC6F6BF79426

TPK$_i$(49octets)=0436399F5558CF34A6E519DA536815252162 16165CFA37A90387068C4
B9F8E1340339F507E999B9F783061A581BBED1116

Sig$_i$(48octets)=AE68EE944D1F3A9DF2F4B1A85A2E0E72585F542E524918AB6C34192DE
D78F890CC7914E2076B5EF5714AE182C2A86EF7

MIC$_i$(32octets)=440D8B4F75E28C6C14FA72612820194B855BFCD154A2224DD55BB239A 260572C

RES$_t$:

Bit 0 to 3(4bits)=0000

Cert$_t$(403octets)=11913082018D30820142A00302010202044A6C47C3300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A130430303033310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D31343130303131363333333325A1 70D3135313031343136333334335A305231143012060A0992268993F
22C6401191604574 15049310B300906035504061302434E310D300B060355040A130430303033310B3009060
355040B1302534E3111300F060355040314087461674041535545304A301406072A86
48C E3D020106092A811CD7630101020103320004A2FA97A41B88FB9552D5CA8EEA325EE51EB
4ADAF884528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A3143012301
006096086480186F842010D0403160178300C06082A811CD763010101050003370030340
218259C42F134F22B96399F13C7C3106A0C103966B0F326384502180EEAC454472711285
6A44F7177D6D1E1BB3A9FEFEFB6C00C

RES$_i$:

Bit 0 to 3(4bits)=0000

Cert$_i$(411octets)=10993082019530820149A00302010202046390D2F5300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A130430303033310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D31343130303131363333333325A1 70D3135313031343136333334335A305931143012060A0992268993F
22C6401191604574 15049310B300906035504061302434E310D300B060355040A130430303033310B3009060
355040B1302534E311830160603550403140F696E746572726F6761746F72404145304A3
01406072A8648CE3D020106092A811CD7630101020103320004BBA79921268948117DA73
45B44479A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A414
29CA3143012301006096086480186F842010D0403160178300C06082A811CD7630101010
50003380030350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F02183
1B27EA8752C2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA

Sig$_{ttp}$(48octets)=B4534F16ED5BE03168C08052E8C97F8B7E1706059C43A3F22E108764
3442D9D03371A1FCFACAA55B49901B9C18C5441F
[MAM1.2 Response]

FN(1octet)=01

RN$_i$(8octets)=8BD3AC6F6BF79426

MIC$_t$(32octets)=42DB9608EB60468799D31FD85076D64A3C3F27E9C7D9E33161B7A0A3C 9459C0E
The final result of authentication process is valid.

D.4 Authenticate_EX

The interrogator sends Authenticate_EX to the TTP and receives the response from the TTP. The Authenticate_EX command and Response are as follows.

TTP Certificate Data:

Cert$_{ttp}$(403octets)=12913082018D30820141A003020102020456692B93300C06082A81 1CD76301010105
00305131143012060A0992268993F22C640119160457415049310B3009 06035504061302434E310D300B060
355040A130430303033310B3009060355040B130253 4E3110300E0603550403140761733140415355301E1
70D3134303130313030303030305A 170D3439313233313233353935395A305131143012060A0992268993F
22C640119160457 415049310B300906035504061302434E310D300B060355040A130430303033310B300906
0355040B1302534E3110300E0603550403140761733140415355304A301406072A864
8CE 3D020106092A811CD76301010201033200049976B163C7BE616FFB84CF05FEB570C2B91D
5270D298FD83012CCF376A217003BA25C6EC6F223E98CEAA53CDB7851F6BA31430123010
06096086480186F842010D0403160178300C06082A811CD7630101010500033800303502
181658297EDA434FBC8A98B69633ADFF52E0F311BA053CBCF3021900902537ED72630A07
4D3DC5FE6B051793308744551C287CF5

[Authenticate_EX]

Cert$_t$(403octets)=11913082018D30820142A00302010202044A6C47C3300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A130430303033310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1

70D31343130303131363333333325A1 70D313531303134313633333334335A305231143012060A0992268993F
22C6401191604574 15049310B300906035504061302434E310D300B060355040A13043030303310B3009060
355040B1302534E3111300F060355040314087461674041535545304A301406072A86
48C E3D020106092A811CD7630101020103320004A2FA97A41B88FB9552D5CA8EEA325EE51EB
4ADAF884528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A3143012301
006096086480186F842010D0403160178300C06082A811CD763010101050003370030340
218259C42F134F22B96399F13C7C3106A0C103966B0F326384502180EEAC454472711285
6A44F7177D6D1E1BB3A9FEFEFB6C00C

Cert$_i$(411octets)=10993082019530820149A00302010202046390D2F5300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A13043030303310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D31343130303131363333333325A1 70D313531303134313633333334335A305931143012060A0992268993F
22C6401191604574 15049310B300906035504061302434E310D300B060355040A13043030303310B3009060
355040B1302534E311830160603550403140F696E746572726F6761746F72404145304A3
01406072A8648CE3D020106092A811CD7630101020103320004BBA79921268948117DA73
45B44479A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A414
29CA3143012301006096086480186F842010D0403160178300C06082A811CD7630101010
50003380030350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F02183
1B27EA8752C2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA

TTPID(8octets)=0031143012060A0992

RN$_t$(8octets)=53CC8496DF6878A4

RN$_i$(8octets)=8BD3AC6F6BF79426

[Authenticate_EX Response]

RN$_t$(8octets)=53CC8496DF6878A4

RN$_i$(8octets)=8BD3AC6F6BF79426

RES$_t$:

Bit 0 to 3(4bits)=0000

Cert$_t$(403octets)=11913082018D30820142A00302010202044A6C47C3300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A13043030303310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D31343130303131363333333325A1 70D313531303134313633333334335A305231143012060A0992268993F
22C6401191604574 15049310B300906035504061302434E310D300B060355040A13043030303310B3009060
355040B1302534E3111300F060355040314087461674041535545304A301406072A86
48C E3D020106092A811CD7630101020103320004A2FA97A41B88FB9552D5CA8EEA325EE51EB
4ADAF884528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A3143012301
006096086480186F842010D0403160178300C06082A811CD763010101050003370030340
218259C42F134F22B96399F13C7C3106A0C103966B0F326384502180EEAC454472711285
6A44F7177D6D1E1BB3A9FEFEFB6C00C

RES$_i$:

Bit 0 to 3(4bits)=0000

Certi(411octets)=10993082019530820149A00302010202046390D2F5300C06082A811 CD763010101050
0305131143012060A0992268993F22C640119160457415049310B30090 6035504061302434E310D300B060
355040A13043030303310B3009060355040B1302534 E3110300E0603550403140761733140415355301E1
70D31343130303131363333333325A1 70D313531303134313633333334335A305931143012060A0992268993F
22C6401191604574 15049310B300906035504061302434E310D300B060355040A13043030303310B3009060
355040B1302534E311830160603550403140F696E746572726F6761746F72404145304A3
01406072A8648CE3D020106092A811CD7630101020103320004BBA79921268948117DA73
45B44479A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A414
29CA3143012301006096086480186F842010D0403160178300C06082A811CD7630101010
50003380030350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F02183
1B27EA8752C2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA

Sig$_{ttp}$(48octets)=B4534F16ED5BE03168C08052E8C97F8B7E1706059C43A3F22E108764
3442D9D03371A1FCFACAA55B49901B9C18C5441F

Annex E
(normative)

Protocol specific operation

E.1 Protocol specific operation

The ECDSA-ECDH Crypto Suite may be used as one of security services for ISO/IEC 18000-4:2018. According to ISO/IEC 29167-1, the Crypto Suite Identifier (CSI) for ECDSA-ECDH crypto suite specified in this document shall be 000110_2, it expands to a 8-bit value 06_h for use by all air interface protocols in this Annex.

ISO/IEC 18000-4:2018, Mode 4

E.1.1 General

ISO/IEC 18000-4:2018 Mode 4 does provide general commands for communication. In order to implement the commands as required according ISO/IEC 18000-4:2018, Clause 9, the commands defined in the additional subclauses in this Annex shall be implemented as part of the payload of the ISO/IEC 18000-4:2018 Mode 4 communication.

E.1.2 Authenticate command

E.1.2.1 Command

E.1.2.1.1 Payload

Payload shall be as shown in Table E.1.

Table E.1 — Payload

Device Type	0x00	Data	CRC[2]

E.1.2.1.2 Data

Data shall be as shown in Table E.2.

Table E.2 — Data

	Command	RFU	CSI	Length	Message	RN	CRC-16
# of bits	8	8	8	EBV	Variable	16	16
description	80_h	00_h	06_h	length of message	message	handle	CRC-16

E.1.2.2 Reply

E.1.2.2.1 Payload

Payload shall be as shown in Table E.3.

Table E.3 — Payload

Device Type	0x01	Data	CRC[2]

E.1.2.2.2 Data

Data shall be as shown in Table E.4.

Table E.4 — Data

	Header	Length	Response	RN	CRC-16
# of bits	1	EBV	Variable	16	16
description	0	length of response	response	handle	CRC-16

E.1.2.3 Authenticate (MAM1.1 Message)

MAM1.1 Message shall be as shown in Table E.5.

Table E.5 — MAM1.1 Message

	Message						
	FN	IID	AuthType	AuthStep	TTPID	Cert$_i$	ECDHP
# of bits	8	64	2	3	Variable	Variable	256
description	fragmentation number	interrogator identifier	00	000	TTP involved or not	digital certificate of interrogator	ECDH parameter

E.1.2.4 MAM1.1 Response

The response of MAM1.1 shall be as shown in Table E.6.

Table E.6 — MAM1.1 Response format

	Response							
	FN	TID	TTPID	Cert$_t$	RN$_t$	TPK$_t$	ECDHP	Sig$_t$
# of bits	8	64	Variable	Variable	64	392	256	384
description	fragme-ntation number	tag identifier	TTP involved or not	digital certificate of tag	random number of tag	temporary public key for ECDH	ECDH parameter	ECDSA signature of tag

E.1.2.5 Authenticate (MAM1.2 Message)

MAM1.2 Message shall be as shown in Table E.7.

Table E.7 — MAM1.2 Message

	Message								
	FN	Auth Type	Auth Step	RN$_t$	RN$_i$	TPK$_i$	Sig$_i$	MIC$_i$	AuthRes
# of bits	8	2	3	64	64	392	384	256	Variable
description	fragme-ntation number	00	001	random number of tag	random number of interro-gator	temporary public key for ECDH	Interro-gator ECDSA signature	message Integrity code	result of TTP authentic-cation

E.1.2.6 MAM1.2 response

The response of MAM1.2 shall be as shown in Table E.8.

Table E.8 — MAM1.2 Response format

	Response		
	FN	**RN_i**	**MIC_t**
# of bits	8	64	256
description	fragmentation number	random number of interrogator	message Integrity code

E.1.3 AuthComm command

E.1.3.1 Command

E.1.3.1.1 Payload

Payload shall be as shown in Table E.9.

Table E.9 — Payload

Device Type	0x00	Data	CRC[2]

E.1.3.1.2 Data

Data shall be as shown in Table E.10.

Table E.10 — Data

	Command	CSI	Length	Message	MAC	RN	CRC-16
# of bits	8	8	EBV	Variable	128	16	16
description	81_h	06_h	length of message	message	message authentication code	handle	CRC-16

E.1.3.2 Reply

E.1.3.2.1 Payload

Payload shall be as shown in Table E.11.

Table E.11 — Payload

Device Type	0x01	Data	CRC[2]

E.1.3.2.2 Data

Data shall be as shown in Table E.12.

Table E.12 — Data

	Header	Length	Response	MAC	RN	CRC-16
# of bits	1	EBV	Variable	128	16	16
description	0	length of response	response	message authentication code	handle	CRC-16

E.1.4 SecureComm command

E.1.4.1 Command

E.1.4.1.1 Payload

Payload shall be as shown in Table E.13.

Table E.13 — Payload

Device Type	0x00	Data	CRC[2]

E.1.4.1.2 Data

Data shall be as shown in Table E.14.

Table E.14 — Data

	Command	CSI	Length	Message	MAC	RN	CRC-16
# of bits	8	8	EBV	Variable	128	16	16
description	82_h	06_h	length of message	encrypted message	message authentication code	handle	CRC-16

E.1.4.2 Reply

E.1.4.2.1 Payload

Payload shall be as shown in Table E.15.

Table E.15 — Payload

Device Type	0x01	Data	CRC[2]

E.1.4.2.2 Data

Data shall be as shown in Table E.16.

Table E.16 — Data

	Header	Length	Response	MAC	RN	CRC-16
# of bits	1	EBV	Variable	128	16	16
description	0	length of response	response	message authentication code	handle	CRC-16

Annex F
(normative)

Protocol message's fragmentation and defragmentation

When the interrogator or tag has an authentication protocol message to send, it needs to get the current MTU size. Comparing the message length with the MTU size, if the message length is larger, the sender shall partition the message into smaller fragments. This is a fragmentation process. Each fragment shall have the same length (except the last one) and a separate authentication protocol header. The length should be an integral multiple of 8 octets with a maximum value smaller than the MTU size.

When these fragments reach the targeted receiver, it is necessary to reassemble them. The information contained in the authentication protocol message fragment header is enough for the receiver to correctly reassemble these fragments into a message.

Each fragment contains information to allow the complete message to be reassembled from its constituent fragments. The authentication protocol header of each fragment contains the information that is used by the destination device to reassemble the message. This is a defragmentation process.

The authentication protocol utilizes FN field in the protocol message header to perform the fragmentation and defragmentation process. For each authentication protocol message from a sender, its FN field includes a unique value (the sequence number of the first authentication protocol message is 1, and that of the following is increased by a degree of 1). Bit 0 of the FN field in the fragment specifies whether or not a fragment is following. Only the last fragment of the message shall have this bit set to 0; all other fragments of the message shall have this bit set to 1.

The timeout-based retransmission mechanism of each authentication protocol message has not been defined in the design of the authentication protocol exchange in this document.

Annex G
(informative)

Examples of ECC parameters

G.1 Principle

Cryptographic algorithms and ECC parameters to be applied to information security mechanism can be subject to national and regional regulations.

G.2 ECC parameters used in the authentication mechanism

In this document, ECC parameters are instantiated and all kind of ECC parameters are supported by the authentication mechanism. ECC parameters to be applied in the authentication mechanism can be subject to national and regional regulations.

Annex H
(normative)

TTP involving

H.1 General

The case of TTP involving in this document is specified for 18000-4:2018, Mode 4.

When both the tag and interrogator contain bit [7:0] value of 0000 0001 in the TTPID field, the TTP shall be involved in the authentication procedure. The optional field AuthRes in MAM 1.2 Message shall be present. The Authenticate_EX and the related Response commands between the TTP and interrogator shall also be performed.

H.2 Requirements

When requiring a TTP to provide authentication service, the entity authentication mechanism specified in ISO/IEC 9798-3 shall apply, the tag and the interrogator shall have the TTP's certificate Cert_{ttp} or public key Q_{ttp}. The interrogator shall know if the TTP is available before performing this document.

H.3 Protocol flows between TTP and interrogator

H.3.1 Authenticate_EX

An Interrogator and a TTP shall implement the Authenticate_EX command; if they do, they shall implement it as shown in Table H.1.

Table H.1 — Authenticate_EX command

	Cert_t	Cert_i	TTPID	RN_t	RN_i
# of bits	Variable	Variable	72	64	64
description	digital certificate of tag	digital certificate of interrogator	identifier of TTP	random number of tag	random number of interrogator

The fields of Authenticate_EX command shall have the following meaning:

a) Cert_t: This field specifies the digital certificate of tag.

b) Cert_i: This field specifies the digital certificate of interrogator.

c) TTPID: This field shall be 72 bits in length and bit [71:8] specifies the TTP identifier. The value is the same as the one in the MAM1.1 Message.

d) RN_t: This field shall be 64 bits in length and specifies the random number generated by the tag. The value is the same as the one in the MAM1.1 Response.

e) RN_i: This field shall be 64 bits in length and specifies the random number generated by the interrogator. The value is the same as the one in the MAM1.2 Message.

H.3.2 Authenticate_EX Response

The response of Authenticate_EX command is as shown in Table H.2

Table H.2 — Authenticate_EX Response

	RN_t	RN_i	RES_t	RES_i	Sig_{ttp}
# of bits	64	64	Variable	Variable	384
description	random number of tag	random number of interrogator	authenticate result of tag	authenticate result of interrogator	TTP ECDSA signature

The fields of Authenticate_EX Response shall have the following meaning:

a) RN_t: This field shall be 64 bits in length and specifies the random number generated by the tag. The value is the same as the one in the Authenticate_EX command.

b) RN_i: This field shall be 64 bits in length and specifies the random number generated by the interrogator. The value is the same as the one in the Authenticate_EX command.

c) RES_t: This field specifies the authenticate result of tag. The first part (Bit [3:0]) is the Flag of the certificates status verification result; the second part is the digital certificate of tag per 7.2. The values of the Flag are as follows:

   1) 0000: the certificate is valid;

   2) 0001: the certificate is invalid;

   3) all other values are RFU.

d) RES_i: This field specifies the authenticate result of interrogator. The first part (Bit [3:0]) is the Flag of the certificates status verification result; the second part is the digital certificate of interrogator per 7.2. The values of the Flag are as follows:

   1) 0000: the certificate is valid;

   2) 0001: the certificate is invalid;

   3) All other values are RFU.

Sig_{ttp}: This field shall be 384 bits in length and specifies the digital signature generated by the TTP. The value is computed by Sig_{ttp}=ECDSA (S_{ttp}, $RN_t||RN_i||RES_t||RES_i$).

H.4 Procedure

When the interrogator received the MAM 1.1 Response, it shall send the Authenticate_EX command to the TTP, including the information of $Cert_t$, $Cert_i$,TTPID,RN_t and RN_i.

After receiving the Authenticate_EX command, the TTP shall verify the tag and interrogator certificate (See RFC 3280 for the certificates status verification) and send the Authenticate_EX Response to the interrogator, including the information of RN_t,RN_i, RES_t,RES_i and Sig_{ttp}. This Authenticate_EX Response contains the identity authentication results of both the tag and interrogator.

When the interrogator received the Authenticate_EX Response from the TTP, it shall check the tag validity by RN_t, $Cert_t$ and the authentication results of the tag RES_t. If the verification is failed, ignore the response and the authentication procedure is failed. Otherwise, send the Authenticate (the optional field AuthRes presents) of MAM 1.2 Message to the tag.

After receiving the Authenticate(MAM 1.2 Message), the tag shall check the interrogator validity by RN_i, $Cert_i$ and the authentication results of the interrogator RES_i. If the verification is failed, then ignore the response; the authentication procedure is failed. Otherwise, send the MAM 1.2 Response to the interrogator.

Bibliography

[1] ISO/IEC 19823-16:2020, *Information technology — Conformance test methods for security service crypto suites — Part 16: Crypto suite ECDSA-ECDH security services for air interface communications*

[2] RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

ICS 35.040.50

Price based on 31 pages