INTERNATIONAL STANDARD

ISO/IEC 29167-16

First edition
2015-11-15

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 par suite cryptographique ECDSA-ECDH pour communications d'interface radio

Reference number
ISO/IEC 29167-16:2015(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 ..... **iv**

Introduction ..... **v**

1 Scope **1**

2 Conformance **1**
2.1 Claiming conformance 1
2.2 Interrogator conformance and obligations 1
2.3 Tag conformance and obligations 2

3 Normative references **2**

4 Terms and definitions **2**

5 Symbols and abbreviated **3**
5.1 Symbols 3
5.2 Abbreviated terms 3

6 Cipher introduction **4**

7 Parameter definitions **4**
7.1 Parameter definitions 4
7.2 Certiticate format 5

8 State diagram **6**

9 Initialization and resetting **6**

10 Authentication **6**
10.1 General 6
10.2 Authenticate message 7
10.2.1 Message in Authenticate command and reply 7
10.2.2 Authenticate(MAM1.1 Message) 8
10.2.3 MAM1.1 Response 8
10.2.4 Authenticate(MAM1.2 Message) 9
10.2.5 MAM1.2 Response 10
10.3 Authentication procedure 11
10.3.1 Protocol requirements 11
10.3.2 Procedure 11

11 Communication **12**
11.1 Authenticate Communication 12
11.2 Secure Communication 13

Annex A (normative) **State transition table** **15**

Annex B (normative) **Error codes and error handling** **16**

Annex C (normative) **Cipher description** **17**

Annex D (informative) **Test Vectors** **18**

Annex E (normative) **Protocol specific** **23**

Annex F (normative) **Protocol message's fragmentation and defragmentation** **28**

Annex G (informative) **Examples of ECC parameters** **29**

Annex H (normative) **TTP involving** **30**

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*, Subcommittee 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: Air Interface for security services and file management for RFID architecture*

— *Part 10: Air Interface for security services crypto suite AES128*

— *Part 11: Air Interface for security services crypto suite PRESENT-80*

— *Part 12: Air Interface for security services crypto suite ECC-DH*

— *Part 13: Air Interface for security services crypto suite Grain-128A*

— *Part 14: Air Interface for security services crypto suite AES-OFB*

— *Part 15: Air Interface for security services crypto suite XOR*

— *Part 16: Air Interface for security services crypto suite ECDSA-ECDH*

— *Part 17: Air Interface for security services crypto suite Crypto GPS*

— *Part 19: Air Interface for security services crypto suite RAMON*

Introduction

This international standard describes a crypto suite based on Elliptic Curve Cryptography (ECC) for the ISO/IEC 18000- series of standards protocol. In particular, it 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 international standard defines only mutual authentication for the ECDSA-ECDH cipher. An Interrogator or a Tag authentication is not supported in this international standard.

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

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 may 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 ensured 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 may be obtained from:

NXP B.V. **411 East Plumeria, San José, CA 95134-1924 USA**
China IWNCOMM Co., LTD. **A201, QinFeng Ge, Xi'an Software Park, No.68 Keji 2nd Road, Xi'an Hi-tech Industrial Development Zone, Shaanxi, P. R. China 710075**
Impinj, Inc. **701 N 34th Street, Suite 300, Seattle, WA 98103 USA**

The latest information on IP that may 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 16:
Crypto suite ECDSA-ECDH security services for air interface communications

1 Scope

This international standard describes a crypto suite based on Elliptic Curve Cryptography (ECC) for the ISO/IEC 18000- series of standards protocol. In particular, it 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 international standard 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 international standard defines a mutual authentication method and methods of use for the cipher. A Tag and an Interrogator may support one, a subset, or all of the specified options, clearly stating what is supported. Key update is not supported in this international standard.

2 Conformance

2.1 Claiming conformance

To claim conformance with this part of ISO/IEC 29167, an Interrogator or a 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 messages and responses format 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 may

— implement any subset of the optional parameters for message and response format defined in this part of ISO/IEC 29167

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

— implement any messages and responses format that conflicts with this part of ISO/IEC 29167, or

— require the use of an optional, proprietary, or custom parameters for message and response format 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 message and response formatting 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 may

— implement any subset of the optional parameters in the message and response formatting defined in this part of ISO/IEC 29167

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

— implement any message and response formatting that conflicts with this part of ISO/IEC 29167, or

— require the use of an optional, proprietary, or custom parameter in the message and response formatting 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 18000-4, *Information technology — Radio frequency identification for item management — Part 4: Parameters for air interface communications at 2,45 GHz*

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*

ISO/IEC 14888-3:2006, *Information technology — Security techniques — Digital signatures with appendix — Part 3: Discrete logarithm based mechanisms*

ISO/IEC 11770-3:2008, *Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques*

ISO/IEC 9798-3:1998/Amd.1:2010, *Information technology — Security techniques — Entity authentication — Part 3: Mechanisms using digital signature techniques / Amendment 1: .*

ISO/IEC 18031:2011, *Information technology — Security techniques — Random bit generation*

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

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

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
command (message)
command that Interrogator sends to Tag with “Message” as parameter

4.2
message
part of the Command that is defined by the CS

4.3
reply (response)
reply that Tag returns to the Interrogator with “Response” as parameter

4.4
response
part of the Reply (stored or sent) that is defined by the CS

5 Symbols and abbreviated

5.1 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
•	Point multiply

5.2 Abbreviated terms

CRC	Cyclic Redundancy Check
CS	Crypto Suite
CSI	Cryptographic Suite Identifier
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
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

6 Cipher introduction

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of the Digital Signature Algorithm (DSA) which uses Elliptic Curve Cryptography (ECC). ECDSA supports mutual authentication and has been specified in ISO/IEC 14888-3.

Elliptic curve Diffie–Hellman (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 has been specified in ISO/IEC 11770-3.

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 — Definition 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)

Table 1 *(continued)*

Parameter	Description
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) 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.
$Cert_x$[Variable]	The digital certificate of x. x can be tag, interrogator or TTP. See 7.2.
RN_t[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.
TPK_t[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 $TPK_t = X_t \bullet P$.
TTPID[Variable]	Specifying whether or not the TTP is to be involved and the identifier of the TTP
Sig_t[383:0]	Digital signature generated by the tag.
RN_i[63:0]	64-bit random number generated by the interrogator.
X_i[391:0]	Temporary private key generated by interrogator and used for ECDH exchange.
TPK_i[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 $TPK_i = X_i \bullet P$.
MIC_i[255:0]	Message integrity code generated by the interrogator.
Sig_i[383:0]	Digital signature generated by the interrogator.
MIC_t[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 RES_t, RES_i and Sig_{ttp}.

7.2 Certificate 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

1. 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.

2. 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. See Annex A.

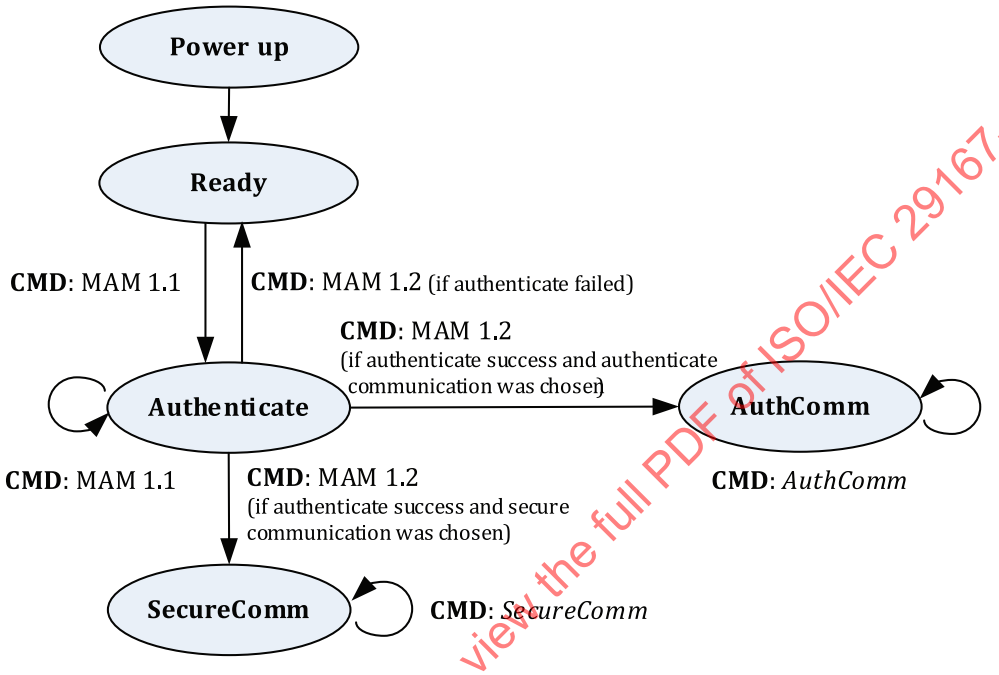

Figure 2 — State diagram

9 Initialization and resetting

This part of ISO/IEC 29167 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 ensure that all memory used for intermediate results is cleared after each operation (message-response pair) and after reset.

10 Authentication

10.1 General

This part of the standard describes additions to the ISO/IEC 18000 series of standards protocol to support the tag and reader air interface security (TRAIS) based on public key cryptography (TRAIS-P). Especially, it defines

1. the use of ECC certificates and Elliptic Curve Digital Signature Algorithm (ECDSA) for mutual authentication of an interrogator and a tag, and

2. the use of the Elliptic Curve Diffie-Hellman (ECDH) key agreement scheme with keys to establish the secure channel, and

3. 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.

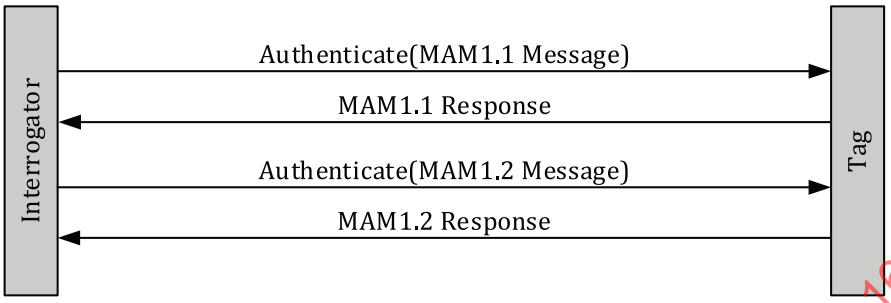

Figure 3 — Mutual authentication with key agreement

The mechanism is based on the ISO/IEC 9798-3. According to ISO/IEC 9798-3:1998/Amd.1:2010, the interrogator and tag can also involve an online trusted third party for the mutual authentication, Figure 4 shows protocol flows between online trusted third party and an interrogator (See Annex H for the case of TTP involving).

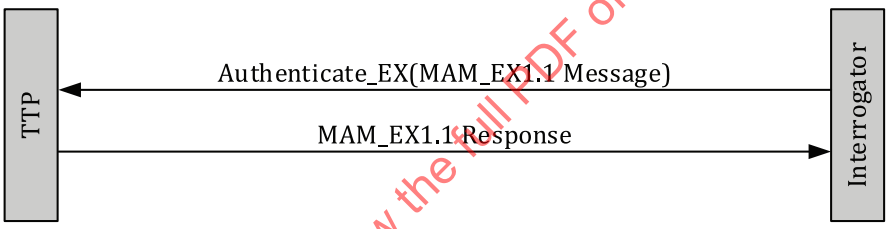

Figure 4 — Protocol flows between TTP and interrogator

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	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

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 (See Annex E).

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 is to 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 (See Annex H for the case of TTP involving). 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 generated by tag	temporary public key generated by tag and used for ECDH exchange	ECDH Parameter	tag ECDSA signature

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 (See Annex E).

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. See Annex H for the case of TTP involving.

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 has been specified in ISO/IEC 18031:2011.

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 $TPK_t = X_t \bullet P$.

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 Sig_t = ECDSA (S_t,TID||IID||$Cert_t$||TTPID||RN_t||TPK_t||ECDHP).

Note This document describes ECC-192. The use of other ECC curves may require an update of the length of command parameters.

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**
de-scrip-tion	fragmen-tation number	00	001	random number gener-ated by tag	random number gener-ated by interro-gator	temporary public key generated by Inter-rogator and used for ECDH exchange	inter-rogator ECDSA signa-ture	message Integrity code gen-erated by Interro-gator	results of TTP authen-tication

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 (See Annex E).

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 meaning:

   — 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 has been specified in ISO/IEC 18031:2011.

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-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. See Annex H for the case of TTP involving.

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 generated by 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 (See Annex E).

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 9798-3:1998, 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. Otherwise, go to step 2).

   2) The interrogator generates temporary private key X_i and temporary public key TPK_i, where $TPK_i = X_i \bullet P$. and uses X_i and TPK_t to perform the ECDH computation, and gets the primary key seed $(X_i \bullet TPK_t)_{abscissa}$, where $(X_i \bullet TPK_t)_{abscissa}$ indecates X coordinate of $X_i \bullet TPK_t$, and $X_i \bullet TPK_t$ shall not be infinite point. The interrogator computes KD-HMAC-SHA256($(X_i \bullet TPK_t)$ 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 is specified in ISO/IEC 11770-6.

   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 frist part is used as integrity authentication key IAK, the second part is used as session integrity check key SIK, the third part is used as 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, ignores 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).

g) 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, ignores the response, the authentication procedure is failed. Otherwise, the authentication process is successful.

NOTE 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 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.

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

11 Communication

11.1 Authenticate Communication

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

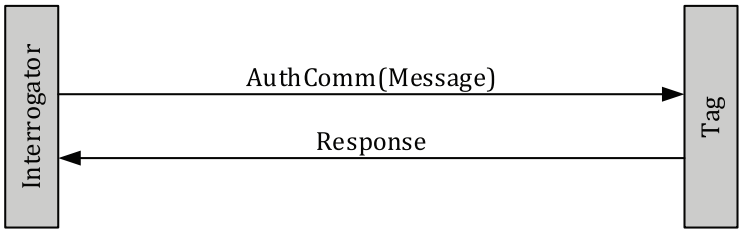

Figure 5 — Authenticate communication

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 message authenticate code of message in AuthComm. The response in reply to an AuthComm command is shown in Table 9, the MAC in response is message authenticate code of Length and Response.

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. Secure Communication uses symmetric encryption algorithms.

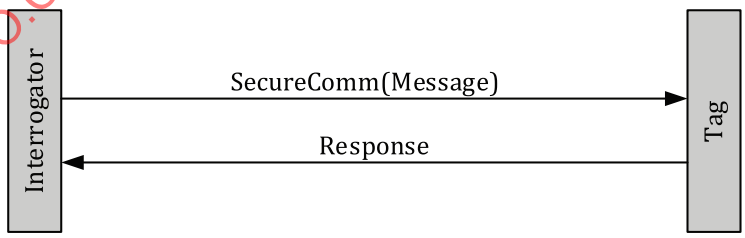

Figure 6 — Secure communication

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 message authenticate code of reponse in SecureComm. The response in reply to a SecureComm command is shown in Table 11 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 an 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 Code	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 code	Description	Error subcode	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.

Annex C
(normative)

Cipher description

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of the Digital Signature Algorithm (DSA) which uses Elliptic Curve Cryptography (ECC), which was specified in ISO/IEC 14888-3. Its security is based on the computational intractability of the discrete logarithm problem.

Elliptic curve Diffie–Hellman (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 International Standard, 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 $x \cdot P$ and $y \cdot P$ are the points in the elliptic curve defined in the elliptic curve domain parameters.

c) The key seed $(x \cdot y \cdot P)_{\text{abscissa}}$ negotiated by ECDH is the x-coordinate at $x \cdot y \cdot P$. $x \cdot y \cdot P$ 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: $Y^2 = X^3 + aX + b$

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 = 8B60CEE3ECBD2AB0F2E00F03C27176767D77B99BB451C47F

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 following, 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 following.

[MAM1.1 Message]

```
FN(1octet)=01
IID(8octets)= 1364615367812479
AuthType(2bits)=00
AuthStep(3bits)=000
TTPID(9octets)=0031143012060A0992
Certi(411octets)=109930820195308201 49A003020102020463 90D2F5300C06082A811
D763010101050003051311430120 60A0992268993F22C6401191604574150493 10B300906
35504061302434E310D300B060355040A1304303030333310B3009060355040B1302534E3
10300E060355040314076173314041535530 1E170D3134313030313136333333325A170D
135313031343136333343 35A30593114301206 0A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A1304303030333310B300906035504
B1302534E311830160603550403140F696E746572726F6761746F72404145304A301406072
2A8648CE3D020106092A811CD763010102010332000 4BBA79921268948117DA7345B4447
A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A41429CA3143
123010060960864801 86F842010D0403160178300C06082A811CD763010101050003380 0
0350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F021831B27EA8752
2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA
ECDHP(32octets)=01000B06092a811cd763010102010000000000000000000000000000000
0000000
```

[MAM1.1 Response]

```
FN(1octet)=01
TID(8octets)= 117A57A1735BB9C6
TTPID(9octets)=0031143012060A0992
CERTt(403octets)=11913082018D30820142A003020102020 44A6C47C3300C06082A811
D763010101050003051311430120 60A0992268993F22C6401191604574150493 10B300906
35504061302434E310D300B060355040A1304303030333310B3009060355040B1302534E3
10300E060355040314076173314041535530 1E170D3134313030313136333333325A170D
135313031343136333343 35A30523114301206 0A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A1304303030333310B300906035504
B1302534E3111300F060355040314086746167404153554530 4A301406072A8648CE3D020
06092A811CD763010102010332 0C4A2FA97A41B88FB9552D5CA8EEA325EE51EB4ADAF88
528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A314301230100609608
480186F842010D0403160178300C06082A811CD763010101050003370030340218259C42
134F22B96399F13C7C3106A0C103966B0F326384502180EEAC4544727112856A44F7177D
D1E1BB3A9FEFEFB6C00C
RNt(8octets)=53CC8496DF6878A4
TPKt(49octets)=048BB733FA69A4EE35D37C9C7205615B0A767BD0B156B6811A8EE73F2
C633F8107245F00960526FEEB10273E5BA5AD290
ECDHP(32octets)=01000B06092a811cd763010102010000000000000000000000000000000
0000000
Sigt(48octets)=B0B294ECC5F1DB9C85B5EBDCF89AA47F359F4526DC33843628786AE35
94DA884C2CF430C2C8D813CE5FF40E949241FA
```

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
RNt(8octets)=53CC8496DF6878A4
RNi(8octets)=8BD3AC6F6BF79426
TPKi(49octets)=0436399F5558CF34A6E519DA5368152521621 6165CFA37A90387068C4
9F8E1340339F507E999B9F783061A581BBED1116
```

```
Sigi(48octets)=AE68EE944D1F3A9DF2F4B1A85A2E0E72585F542E524918AB6C34192DE
78F890CC7914E2076B5EF5714AE182C2A86EF7
MICi(32octets)=440D8B4F75E28C6C14FA72612820194B855BFCD154A2224DD55BB239A
60572C
RESt:
Bit 0 to 3(4bits)=0000
CERTt(403octets)=11913082018D30820142A00302010202044A6C47C3300C06082A811
D763010101050003051311430120 60A0992268993F22C64011916045741504931 0B300906
35504061302434E310D300B060355040A130430303033310B3009060355040B1302534E3
10300E06035504031407617331404153555301E170D313431303031313633333325A170D
135313031343136333334335A305231143012060A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A130430303033310B300906035504
B1302534E3111300F06035504031408746167404153554530 4A301406072A8648CE3D020
06092A811CD763010102010332000 4A2FA97A41B88FB9552D5CA8EEA325EE51EB4ADAF88
528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A314301230100609608
480186F842010D0403160178300C06082A811CD763010101050003370030340218259C42
134F22B96399F13C7C3106A0C103966B0F326384502180EEAC4544727112856A44F7177D
D1E1BB3A9FEFEFB6C00C
RESi:
Bit 0 to 3(4bits)=0000
Certi(411octets)=109930820195308201 49A00302010202046390D2F5300C06082A811
D763010101050003051311430120 60A0992268993F22C64011916045741504931 0B300906
35504061302434E310D300B060355040A130430303033310B3009060355040B1302534E3
10300E06035504031407617331404153555301E170D313431303031313633333325A170D
135313031343136333334335A305931143012060A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A130430303033310B300906035504
B1302534E3118301606035504031 40F696E746572726F6761746F7204145304A3014060
2A8648CE3D020106092A811CD763010102010332000 4BBA799212689481 17DA7345B4447
A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A41429CA3143
123010060960864801 86F842010D0403160178300C06082A811CD763010101050003380 0
0350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F021831B27EA8752
2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA
Sigttp(48octets)=B4534F16ED5BE03168C08052E8C97F8B7E1706059C43A3F22E10876
3442D9D03371A1FCFACAA55B49901B9C18C5441F
```

[MAM1.2 Response]

```
FN(1octet)=01
RNi(8octets)=8BD3AC6F6BF79426
MICt(32octets)=42DB9608EB60468799D31FD85076D64A3C3F27E9C7D9E33161B7A0A3C9459C0E
```

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 following:

TTP Certificate Data:

```
CERTttp(403octets)=12913082018D30820141A00302010202045692B93300C06082A8
1CD763010101050003051311430120 60A0992268993F22C64011916045741504931 0B3009
6035504061302434E310D300B060355040A130430303033310B3009060355040B1302534
3110300E06035504031407617331404153555301E170D313430313031303030303030305A17
D343931323331323335393539 5A305131143012060A0992268993F22C640119160457415
49310B300906035504061302434E310D300B060355040A130430303033310B3009060355
40B1302534E3110300E06035504031407617331404153553 04A301406072A8648CE3D020
06092A811CD763010102010332000 49976B163C7BE616FFB84CF05FEB570C2B91D5270D2
8FD83012CCF376A217003BA25C6EC6F223E98CEAA53CDB7851F6BA314301230100609608
480186F842010D0403160178300C06082A811CD763010101050003380030350218165829
EDA434FBC8A98B69633ADFF52E0F311BA053CBCF3021900902537ED72630A074D3DC5FE6
051793308744551C287CF5
```

[Authenticate_EX]

```
CERTt(403octets)=11913082018D30820142A00302010202044A6C47C3300C06082A811
D763010101050003051311430120 60A0992268993F22C64011916045741504931 0B300906
```

```
3550406130243 4E310D300B060355040A130430303033310B3009060355040B1302534E3
10300E060355040314076173314041535530 1E170D3134313030313136333333325A170D
135313031343136333334335A305231143012060A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A130430303033310B300906035504
B1302534E3111300F060355040314087461674041535545304A301406072A8648CE3D020
06092A811CD7630101020103320004A2FA97A41B88FB9552D5CA8EEA325EE51EB4ADAF88
528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A314301230100609608
480186F842010D0403160178300C06082A811CD7630101010500033700303402182 59C42
134F22B96399F13C7C3106A0C103966B0F326384502180EEAC4544727112856A44F7177D
D1E1BB3A9FEFEFB6C00C
Certi(411octets)=1099308201953082014 9A003020102020 46390D2F5300C06082A811
D7630101010500305131143012060A0992268993F22C64011916045741504 9310B300906
35504061302434E310D300B060355040A130430303033310B3009060355040B1302534E3
10300E060355040314076173314041535530 1E170D3134313030313136333333325A170D
135313031343136333334335A305931143012060A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A130430303033310B300906035504
B1302534E3118301606035504031 40F696E746572726F6761746F72404145304A3014060
2A8648CE3D020106092A811CD7630101020103320004BBA79921268948117DA7345B4447
A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A41429CA3143
123010060960864801 86F842010D0403160178300C06082A811CD7630101010500033800
0350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F021831B27EA8752
2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA
TTPID(8octets)=0031143012060A0992
RNt(8octets)=53CC8496DF6878A4
RNi(8octets)=8BD3AC6F6BF79426
```

[Authenticate_EX Response]

```
RNt(8octets)=53CC8496DF6878A4
RNi(8octets)=8BD3AC6F6BF79426
RESt:
Bit 0 to 3(4bits)=0000
CERTt(403octets)=11913082018D30820142A003020102020 44A6C47C3300C06082A811
D7630101010500305131143012060A0992268993F22C64011916045741504 9310B300906
35504061302434E310D300B060355040A130430303033310B3009060355040B1302534E3
10300E060355040314076173314041535530 1E170D3134313030313136333333325A170D
135313031343136333334335A305231143012060A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A130430303033310B300906035504
B1302534E3111300F060355040314087461674041535545304A301406072A8648CE3D020
06092A811CD7630101020103320004A2FA97A41B88FB9552D5CA8EEA325EE51EB4ADAF88
528013CC11046A795F340E38F7EEB4B2BF00AC3195D60C0482878A314301230100609608
480186F842010D0403160178300C06082A811CD7630101010500033700303402182 59C42
134F22B96399F13C7C3106A0C103966B0F326384502180EEAC4544727112856A44F7177D
D1E1BB3A9FEFEFB6C00C
RESi:
Bit 0 to 3(4bits)=0000
Certi(411octets)=1099308201953082014 9A003020102020 46390D2F5300C06082A811
D7630101010500305131143012060A0992268993F22C64011916045741504 9310B300906
35504061302434E310D300B060355040A130430303033310B3009060355040B1302534E3
10300E060355040314076173314041535530 1E170D3134313030313136333333325A170D
135313031343136333334335A305931143012060A0992268993F22C64011916045741504
310B300906035504061302434E310D300B060355040A130430303033310B300906035504
B1302534E3118301606035504031 40F696E746572726F6761746F72404145304A3014060
2A8648CE3D020106092A811CD7630101020103320004BBA79921268948117DA7345B4447
A3EBDF7FE54EE26EBF1A36C1B8E9FF1B7E4EDDBA574BB34C795B0599CFF7A41429CA3143
123010060960864801 86F842010D0403160178300C06082A811CD7630101010500033800
0350219008FAB674B943BF646D6D27A70D3DDE435B3C46F6530D8839F021831B27EA8752
2F2DC5CB22951ECEDE8C0FEAD4A02B96BCEA
Sigttp(48octets)=B4534F16ED5BE03168C08052E8C97F8B7E1706059C43A3F22E10876
3442D9D03371A1FCFACAA55B49901B9C18C5441F
```

Annex E
(normative)

Protocol specific

E.1 Protocol Specific Operation

The ECDSA-ECDH Crypto Suite may be used as one of security services for ISO/IEC 18000-4. ISO/IEC 29167-1 defines the Crypto Suite Identifier (CSI) for ECDSA-ECDH to be 000110_2 and it is expanded to the 8-bit value 06_h for use by all air interface protocols in this Annex.

E.2 ISO/IEC 18000-4 Mode 3

E.2.1 General

ISO/IEC 18000-4 Mode 3 does provide general commands for communication. In order to implement the commands as required according chapter 8 of this document these commands shall be implemented as part of the payload of the ISO/IEC 18000-4 Mode 3 communication as defined in the additional sub clauses in this Annex.

E.2.2 Authenticate command

E.2.2.1 Command:

Payload:

Payload shall be as shown in Table E.1.

Table E.1 — Payload

Device Type	0x00	Data	CRC[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	01_h	00_h	06_h	length of message	message	handle	CRC-16

E.2.2.2 Reply

Payload:

Payload shall be as shown in Table E.3.

Table E.3 — Payload

Device Type	0x01	Data	CRC[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.2.2.3 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.2.2.4 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	fragmentation number	Tag identifier	TTP involved or not	Digital certificate of tag	random number generated by tag	temporary public key generated by tag and used for ECDH exchange	ECDH Parameter	tag ECDSA signature

E.2.2.5 Authenticate(MAM1.2 Message)

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