

**Standard** ECMA-410

2<sup>nd</sup> Edition / June 2015

**NFC-SEC-03:  
NFC-SEC Entity  
Authentication and Key  
Agreement using  
Asymmetric  
Cryptography**

**Standard**



**COPYRIGHT PROTECTED DOCUMENT**

| <b>Contents</b>            |   | <b>Page</b> |
|----------------------------|---|-------------|
| <b>1</b>                   | <b>Scope .....</b>                                  | <b>1</b>    |
| <b>2</b>                   | <b>Conformance .....</b>                            | <b>1</b>    |
| <b>3</b>                   | <b>Normative references .....</b>                   | <b>1</b>    |
| <b>4</b>                   | <b>Terms and definitions .....</b>                  | <b>2</b>    |
| <b>5</b>                   | <b>Conventions and notations .....</b>              | <b>3</b>    |
| <b>6</b>                   | <b>Acronyms .....</b>                               | <b>3</b>    |
| <b>7</b>                   | <b>General .....</b>                                | <b>4</b>    |
| <b>8</b>                   | <b>Fields and PDUs for NEAU-A .....</b>             | <b>5</b>    |
| 8.1                        | Protocol Identifier (PID) .....                     | 5           |
| 8.2                        | NFC-SEC-PDUs .....                                  | 5           |
| 8.3                        | TTP involving .....                                 | 6           |
| 8.3.1                      | TTP policy and field .....                          | 6           |
| 8.3.2                      | TTP policy negotiation .....                        | 6           |
| 8.4                        | Entity identifiers .....                            | 7           |
| 8.5                        | Cert field .....                                    | 7           |
| 8.6                        | Res field .....                                     | 7           |
| <b>9</b>                   | <b>Primitives .....</b>                             | <b>8</b>    |
| 9.1                        | General requirements .....                          | 8           |
| 9.2                        | Entity authentication .....                         | 9           |
| 9.2.1                      | Mechanisms .....                                    | 9           |
| 9.2.2                      | EC curve .....                                      | 10          |
| 9.2.3                      | ECDSA .....   | 10          |
| 9.2.4                      | Certificate validation .....                        | 12          |
| 9.3                        | Key agreement .....                                 | 12          |
| 9.4                        | Key confirmation .....                              | 13          |
| 9.5                        | Key Derivation Function (KDF) .....                 | 13          |
| <b>10</b>                  | <b>NEAU-A mechanism .....</b>                       | <b>13</b>   |
| 10.1                       | Entity authentication involving a TTP .....         | 13          |
| 10.1.1                     | Protocol overview .....                             | 13          |
| 10.1.2                     | Preparation .....                                   | 13          |
| 10.1.3                     | Sender (A) transformation .....                     | 14          |
| 10.1.4                     | Recipient (B) transformation .....                  | 15          |
| 10.1.5                     | TTP transformation .....                            | 16          |
| 10.2                       | Entity authentication without involving a TTP ..... | 16          |
| 10.2.1                     | Protocol overview .....                             | 16          |
| 10.2.2                     | Preparation .....                                   | 17          |
| 10.2.3                     | Sender (A) transformation .....                     | 17          |
| 10.2.4                     | Recipient (B) transformation .....                  | 18          |
| 10.3                       | Key derivation .....                                | 19          |
| 10.3.1                     | Sender (A) .....                                    | 19          |
| 10.3.2                     | Recipient (B) .....                                 | 19          |
| <b>11</b>                  | <b>Data Authenticated Encryption in SCH .....</b>   | <b>20</b>   |
| <b>Annex A (normative)</b> | <b>UDP Port 5111 and TAEP .....</b>                 | <b>21</b>   |
| A.1                        | UDP and port 5111 .....                             | 21          |
| A.1.1                      | UDP .....   | 21          |
| A.1.2                      | Port 5111 .....                                     | 21          |
| A.2                        | TAEP .....  | 22          |

|                     |   |           |
|---------------------|---|-----------|
| <b>A.2.1</b>        | <b>TAEP packet format</b> .....               | <b>22</b> |
| <b>A.2.2</b>        | <b>TAEP_REQ and TAEP_RES format</b> .....     | <b>22</b> |
| <b>Annex B</b>      | <b>(informative) ECDSA test vectors</b> ..... | <b>23</b> |
| <b>Bibliography</b> | .....   | <b>24</b> |

## Introduction

The NFC Security series of standards comprise a common services and protocol Standard and NFC-SEC cryptography standards.

This NFC-SEC cryptography Standard specifies an NFC Entity Authentication (NEAU) mechanism that uses the asymmetric cryptography algorithm (NEAU-A) for mutual authentication of two NFC entities.

This Standard addresses entity authentication of two NFC entities possessing certificates and private keys during key agreement and key confirmation for the Shared Secret Service (SSE) and Secure Channel Service (SCH).

This Standard adds entity authentication to the services provided by ISO/IEC 13157-3 (ECMA-409) NFC-SEC-02.

This 2<sup>nd</sup> edition refers to the latest standards.

This Ecma Standard has been adopted by the General Assembly of June 2015.

**"COPYRIGHT NOTICE**

© 2015 Ecma International

*This document may be copied, published and distributed to others, and certain derivative works of it may be prepared, copied, published, and distributed, in whole or in part, provided that the above copyright notice and this Copyright License and Disclaimer are included on all such copies and derivative works. The only derivative works that are permissible under this Copyright License and Disclaimer are:*

- (i) works which incorporate all or portion of this document for the purpose of providing commentary or explanation (such as an annotated version of the document),*
- (ii) works which incorporate all or portion of this document for the purpose of incorporating features that provide accessibility,*
- (iii) translations of this document into languages other than English and into different formats and*
- (iv) works by making use of this specification in standard conformant products by implementing (e.g. by copy and paste wholly or partly) the functionality therein.*

*However, the content of this document itself may not be modified in any way, including by removing the copyright notice or references to Ecma International, except as required to translate it into languages other than English or into a different format.*

*The official version of an Ecma International document is the English language version on the Ecma International website. In the event of discrepancies between a translated version and the official version, the official version shall govern.*

*The limited permissions granted above are perpetual and will not be revoked by Ecma International or its successors or assigns.*

*This document and the information contained herein is provided on an "AS IS" basis and ECMA INTERNATIONAL DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."*

# NFC-SEC-03: NFC-SEC Entity Authentication and Key Agreement using Asymmetric Cryptography

## 1 Scope

This Standard specifies the message contents and the cryptographic mechanisms for PID 03.

This Standard specifies key agreement and confirmation mechanisms providing mutual authentication, using asymmetric cryptography, and the transport protocol requirements for the exchange between Sender and TTP.

NOTE This Standard adds entity authentication to the services provided by ISO/IEC 13157-3 (ECMA-409) NFC-SEC-02.

## 2 Conformance

Conformant NFC-SEC entities employ the security mechanisms and the transport protocol requirements specified in this NFC-SEC cryptography Standard (identified by PID 03) and conform to ISO/IEC 13157-1 (ECMA-385).

Conformant TTP implementations employ the security mechanisms and the transport protocol requirements specified in this NFC-SEC cryptography Standard (identified by PID 03).

The NFC-SEC security services shall be established through the protocol specified in ISO/IEC 13157-1 (ECMA-385) and the mechanisms specified in this Standard.

## 3 Normative references

The following referenced documents are indispensable for the application 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 7498-1:1994, *Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*

ISO/IEC 9798-1:2010, *Information technology -- Security techniques -- Entity authentication -- Part 1: General*

ISO/IEC 9798-3:1998, *Information technology -- Security techniques -- Entity authentication -- Part 3: Mechanisms using digital signature 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 10118-3:2004, *Information technology -- Security techniques -- Hash-functions -- Part 3: Dedicated hash-functions*

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

ISO/IEC 13157-1, *Information technology -- Telecommunications and information exchange between systems -- NFC Security -- Part 1: NFC-SEC NFCIP-1 security services and protocol (ECMA-385)*

ISO/IEC 13157-2, *Information technology -- Telecommunications and information exchange between systems -- NFC Security -- Part 2: NFC-SEC cryptography standard using ECDH and AES (ECMA-386)*

ISO/IEC 13157-3, *Information technology -- Telecommunications and information exchange between systems -- NFC Security -- Part 3: NFC-SEC Cryptography Standard using ECDH-256 and AES-GCM* (ECMA-409)

ISO/IEC 14443-3, *Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 3: Initialization and anticollision*

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

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

ISO/IEC 18031:2011/Cor.1:2014, *Information technology -- Security techniques -- Random bit generation -- Technical Corrigendum 1*

ISO/IEC 18092, *Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)* (ECMA-340)

ITU-T Recommendation X.509, ISO/IEC 9594-8, *Information technology -- Open Systems Interconnection -- The Directory: Public-key and attribute certificate frameworks.*

## 4 Terms and definitions

Clause 4 of ISO/IEC 13157-3 (ECMA-409) applies. Additionally, the following terms and definitions apply.

**4.1 asymmetric cryptography (asymmetric cryptographic technique)**  
cryptographic technique that uses two related transformations: a public transformation (defined by the public key) and a private transformation (defined by the private key)

NOTE The two transformations have the property that, given the public transformation, it is computationally infeasible to derive the private transformation.

[ISO/IEC 9798-1: 2010]

**4.2 certificate**  
public key information of an entity signed by the certification authority and thereby rendered unforgeable

[ISO/IEC 9798-1: 2010]

**4.3 digital signature (signature)**  
data appended to, or a cryptographic transformation of, a data unit that allows the recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g. by the recipient

[ISO/IEC 9798-1: 2010]

**4.4 entity authentication**  
corroboration that an entity is the one claimed

[ISO/IEC 9798-1: 2010]

**4.5 n-entity-title**  
a name that is used to identify unambiguously an n-entity

[ISO/IEC 7498-1: 1994]



#### 4.6

##### **trusted third party**

security authority or its agent, trusted by other entities with respect to security related activities

[ISO/IEC 9798-1: 2010]

NOTE In this Standard, a trusted third party is trusted by a Sender and Recipient for the purposes of certificate validation.

## 5 Conventions and notations

Clause 5 of ISO/IEC 13157-3 (ECMA-409) applies.

For any message field "F", F denotes the value placed in the field upon sending, F' the value upon receipt.

## 6 Acronyms

Clause 6 of ISO/IEC 13157-3 (ECMA-409) applies. Additionally, the following acronyms apply.

|              |   |
|--------------|---|
| CertA        | Certificate of A  |
| CertB        | Certificate of B  |
| CertTTP      | Certificate of TTP  |
| CPA          | Public Key of Certificate of A                                |
| CPB          | Public Key of Certificate of B                                |
| CPTTP        | Public Key of Certificate of TTP                              |
| CSA          | Private Key corresponding to Certificate of A                 |
| CSB          | Private Key corresponding to Certificate of B                 |
| CSTTP        | Private Key corresponding to Certificate of TTP               |
| Dual_EC_DRBG | Dual Elliptic Curve Deterministic Random Bit Generator        |
| ECDSA        | Elliptic Curve Digital Signature Algorithm                    |
| IP           | Internet Protocol   |
| k            | Fresh random value in ECDSA                                   |
| NEAU         | NFC Entity Authentication                                     |
| NEAU-A       | NEAU using Asymmetric Cryptography                            |
| OCSP         | Online Certificate Status Protocol                            |
| q            | 224-bit prime number of a divisor of the curve order in ECDSA |
| r, s         | Digital Signature value of ECDSA                              |
| ResA         | Validation result of A  |
| ResB         | Validation result of B  |
| SHA          | Secure Hash Algorithm   |
| SigA         | Digital Signature generated by A                              |
| SigB         | Digital Signature generated by B                              |
| SigTTP       | Digital Signature generated by TTP                            |
| TTP PolicyX  | TTP policy of entity X [see 8.3]                              |
| TLV          | Type-length-value   |

|          |  |
|----------|--|
| UDP      | User Datagram Protocol                             |
| UID      | Unique Identifier [ISO/IEC 14443-3]                |
| TAEP     | Tri-element Authentication Extensible Protocol     |
| TAEP_REQ | TAEP Request PDU                                   |
| TAEP_RES | TAEP Response PDU                                  |
| TTP      | Trusted Third Party involved in the authentication |

## 7 General

This Standard specifies the NFC Entity Authentication using Asymmetric cryptography (NEAU-A), using the key agreement and confirmation protocol of ISO/IEC 13157-1 (ECMA-385). NEAU-A specifies negotiation of authentication either involving a TTP per 6.2 of ISO/IEC 9798-3 or without TTP per 5.2.2 of ISO/IEC 9798-3.

Authentication credentials shall be Public Key Certificates conforming to ISO/IEC 9594-8 / ITU X.509.

NOTE It is outside the scope of this Standard how the certificates and the related private keys are issued and established.

The relationship between NEAU-A and ISO/IEC 13157-1 (ECMA-385) is shown in Figure 1.

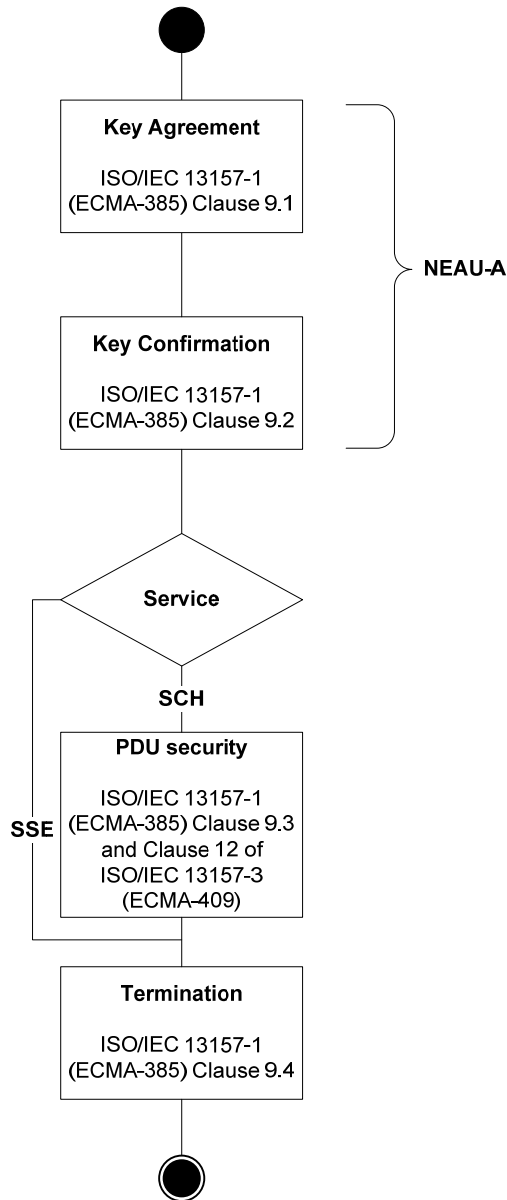


Figure 1 — The use of the NFC-SEC protocol by NEAU-A

## 8 Fields and PDUs for NEAU-A

### 8.1 Protocol Identifier (PID)

This Standard shall use the one octet protocol identifier PID with value 3.

### 8.2 NFC-SEC-PDUs

Peer NFC-SEC entities shall establish a shared secret  $Z$  using ACT\_REQ, ACT\_RES, VFY\_REQ and VFY\_RES according to the NEAU-A mechanism.

### 8.3 TTP involving

#### 8.3.1 TTP policy and field

TTP Policy<sub>x</sub> specifies the entity policy regarding the involvement of the TTP in NEAU-A. The payload of ACT\_REQ and ACT\_RES shall contain the 1-octet TTP field encoding the TTP Policy<sub>x</sub> as follows:

- a) 0: TTP to be involved;
- b) 1: TTP not to be involved;
- c) 2: No preference;
- d) All other values are RFU.

#### 8.3.2 TTP policy negotiation

The NEAU-A mechanism provides a method for TTP policy negotiation. Peer NFC-SEC entities shall negotiate whether or not to involve the TTP, in accordance with their TTP Policy<sub>x</sub>.

The Sender (A) shall include a TTP field in the ACT\_REQ with the value (0, 1 or 2) according to its TTP Policy<sub>A</sub>. If the TTP is unavailable (see 10.1.2) then the values 0 and 2 are prohibited. The value 2 shall be replaced by 1, and if the value is 0 then 'PDU content valid' shall be set to false.

Upon receiving the ACT-REQ, the Recipient (B) shall perform policy negotiation as specified in Table 1; if the Result is False then the Recipient shall set 'PDU content valid' to false, for the Result of 0 or 1, the Recipient (B) shall set the TTP field in the ACT-RES to the Result and shall continue with step 3 of 10.1.4 or step 4 of 10.2.4 respectively.

The Sender (A) shall validate the TTP field in the ACT-RES:

- If it equals 2, then set 'PDU content valid' to false;

Otherwise, evaluate Table 1; if the Result is False then set 'PDU content valid' to false, for the Result of 0 or 1 continue with step 6 of 10.1.3 or 10.2.3 respectively.

**Table 1 — Results of the TTP policy negotiation**

| TTP Field | TTP Policy             | Result |
|-----------|------------------------|--------|
| 0         | TTP to be involved     | 0      |
| 0         | TTP not to be involved | False  |
| 0         | No preference          | 0      |
| 1         | TTP to be involved     | False  |
| 1         | TTP not to be involved | 1      |
| 1         | No preference          | 1      |
| 2         | TTP to be involved     | 0      |
| 2         | TTP not to be involved | 1      |
| 2         | No preference          | 0      |

## 8.4 Entity identifiers

The n-entity-title of the Sender's and Recipient's n-entity shall be used as  $ID_S$  and  $ID_R$ , respectively. Figure 2 specifies the encoding of  $ID_S$  and  $ID_R$  in the TLV format.

|                | ID Type | ID Length | Value    |
|----------------|---------|-----------|----------|
| <b>Octets:</b> | 1       | 2         | variable |

**Figure 2 — ID format**

1. The Type subfield specifies the type of the ID and shall be 1 octet in length. The values are:
  - a) 1: Value subfield contains Sender (A) identification number,  $ID_S$ ;
  - b) 2: Value subfield contains Recipient (B) identification number,  $ID_R$ ;
  - c) All other values are RFU.
2. The 2-octet Length subfield contains the length in number of octets of the Value subfield, in the range of 1 to 65535.

## 8.5 Cert field

Figure 3 specifies the encoding of  $Cert_A$ ,  $Cert_B$  and  $Cert_{TTP}$  in the TLV format.

|                | Cert Type | Cert Length | Value    |
|----------------|-----------|-------------|----------|
| <b>Octets:</b> | 1         | 2           | variable |

**Figure 3 — Cert format**

1. The Cert Type subfield specifies the type of the certificate and shall be 1 octet in length. The values are:
  - a) 0: Value subfield contains certificate of Sender (A),  $Cert_A$ ;
  - b) 1: Value subfield contains certificate of Recipient (B),  $Cert_B$ ;
  - c) 2: Value subfield contains certificate of TTP,  $Cert_{TTP}$ ;
  - d) All other values are RFU.
2. The 2-octet Cert Length subfield contains the length in number of octets of the Value subfield, in the range of 1 to 65535.

## 8.6 Res field

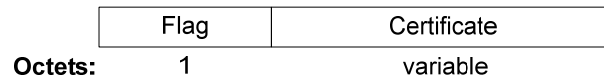
Figure 4 specifies the encoding of  $Res_A$  and  $Res_B$  in the TLV format.

|                | Res Type | Res Length | Value    |
|----------------|----------|------------|----------|
| <b>Octets:</b> | 1        | 2          | variable |

**Figure 4 — Res format**

1. The Res Type subfield identifies the entity and shall be 1 octet in length. The values are:
  - a) 0: result of Sender (A);

- b) 1: result of Recipient (B);
  - c) All other values are RFU.
2. The 2-octet Res Length subfield contains the length in number of octets of the Certificate subfield of the Value subfield, in the range of 1 to 65535.
  3. Figure 5 specifies the Value subfield.



**Figure 5 — Value format**

- a) The Flag subfield contains the certificate validation result, the values are:
  - i. 0: positive;
  - ii. 1: negative;
  - iii. All other values are RFU.
- b) The Certificate subfield contains the certificate verified by the TTP.

## 9 Primitives

### 9.1 General requirements

Clause 9 specifies the cryptographic primitives of NEAU-A. Clause 10 specifies the use of these primitives. Table 2 specifies the size and description of parameters.

**Table 2 — NEAU-A parameters**

| Parameter           | Field Size | Description  |
|---------------------|------------|--|
| Cert <sub>A</sub>   | Variable   | The certificate of A. See 8.5.                           |
| Cert <sub>B</sub>   | Variable   | The certificate of B. See 8.5.                           |
| Cert <sub>TTP</sub> | Variable   | The certificate of TTP. See 8.5.                         |
| CP <sub>A</sub>     | 512 bits   | Public key of certificate of A.                          |
| CP <sub>B</sub>     | 512 bits   | Public key of certificate of B.                          |
| CP <sub>TTP</sub>   | 512 bits   | Public key of certificate of TTP.                        |
| CS <sub>A</sub>     | 256 bits   | Private key corresponding to certificate of A.           |
| CS <sub>B</sub>     | 256 bits   | Private key corresponding to certificate of B.           |
| CS <sub>TTP</sub>   | 256 bits   | Private key corresponding to certificate of TTP.         |
| TTP                 | 8 bits     | This field specifies whether or not the TTP is involved. |
| Sig <sub>A</sub>    | 512 bits   | Digital signature generated by A.                        |

| Parameter          | Field Size | Description   |
|--------------------|------------|---|
| Sig <sub>B</sub>   | 512 bits   | Digital signature generated by B.                           |
| Sig <sub>TTP</sub> | 512 bits   | Digital signature generated by TTP.                         |
| Res <sub>A</sub>   | Variable   | This field specifies the authenticate result of A. See 8.6. |
| Res <sub>B</sub>   | Variable   | This field specifies the authenticate result of B. See 8.6. |
| ID <sub>S</sub>    | Variable   | The Sender (A) identification number. See 8.4.              |
| ID <sub>R</sub>    | Variable   | The Recipient (B) identification number. See 8.4.           |
| NA                 | 128 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| NB                 | 128 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| d <sub>A</sub>     | 256 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| d <sub>B</sub>     | 256 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| Q <sub>A</sub>     | 512 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| Q <sub>B</sub>     | 512 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| QA                 | 264 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| QB                 | 264bits    | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| Z                  | 512 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| MK                 | 128 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |
| K                  | 128 bits   | See Clause 6 of ISO/IEC 13157-2 (ECMA-386).                 |

ISO/IEC 18031 shall be used to generate the random nonces and keys, with the exception of Dual\_EC\_DRBG.

## 9.2 Entity authentication

### 9.2.1 Mechanisms

This Standard specifies two entity authentication options: entity authentication involving a TTP and entity authentication without involving a TTP. Which option to use is negotiated as specified in 8.3.2.

Peer NFC-SEC entities achieve mutual authentication between Sender (A) and Recipient (B) involving a TTP per 6.2 of ISO/IEC 9798-3:1998/Amd.1, Five pass authentication (initiated by A), where the Sender acts as entity A. The entity authentication mechanism requires the two entities, Sender (A) and Recipient (B), to validate each other's certificates using a TTP. To verify the signature of the TTP, the entities Sender (A) and Recipient (B) shall possess a copy of the TTP's certificate.

Peer NFC-SEC entities achieve mutual authentication between Sender (A) and Recipient (B) without involving a TTP per 5.2.2 of ISO/IEC 9798-3:1998, Three pass authentication.

Implementations of this entity authentication mechanism shall use the digital signature scheme Elliptic Curve Digital Signature Algorithm (ECDSA) and control the uniqueness/timeliness by generating and checking a random number (see ISO/IEC 9798-1 and ISO/IEC 18031).

## 9.2.2 EC curve

EC curve P-256 per ISO/IEC 13157-3 (ECMA-409) shall be used for ECDSA and Elliptic Curves Diffie-Hellman (ECDH).

## 9.2.3 ECDSA

### 9.2.3.1 Overview

The digital signature scheme ECDSA used in the entity authentication mechanism shall conform to 6.4 of ISO/IEC 14888-3. The public verification keys used in the ECDSA shall be Public key of certificate of Sender  $CP_A$ , Public key of certificate of Recipient  $CP_B$  and Public key of certificate of TTP  $CP_{TTP}$  respectively. The secret signature keys used in the ECDSA shall be Private key corresponding to certificate of Sender  $CS_A$ , Private key of certificate corresponding to Recipient  $CS_B$  and Private key corresponding to certificate of TTP  $CS_{TTP}$ , respectively.

### 9.2.3.2 Digital signature generation

#### 9.2.3.2.1 Digital signature generation of Sig<sub>A</sub> in VFY\_REQ

The ECDSA digital signature Sig<sub>A</sub> in VFY\_REQ shall be generated per 6.4.3 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of A,
2. The secret signature key  $CS_A$ ,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for signing:  $ID_S || ID_R || NA || NB' || QA$ , and
5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

#### 9.2.3.2.2 Digital signature generation of Sig<sub>B</sub> in ACT\_RES

The ECDSA digital signature Sig<sub>B</sub> in ACT\_RES shall be generated per 6.4.3 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of B,
2. The secret signature key  $CS_B$ ,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for signing:  $ID_R || ID_S || TTP || NB || NA' || QB$ , and
5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

#### 9.2.3.2.3 Digital signature generation of Sig<sub>TTP</sub> in TAEP\_RES

In the case of entity authentication involving a TTP, the ECDSA digital signature Sig<sub>TTP</sub> in TAEP\_RES shall be generated per 6.4.3 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of TTP,
2. The secret signature key  $CS_{TTP}$ ,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for signing:  $NA' || NB' || Res_A || Res_B$ , and



5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

### **9.2.3.3 Digital signature verification**

#### **9.2.3.3.1 Digital signature verification of Sig<sub>A</sub> in VFY\_REQ**

The ECDSA digital signature Sig<sub>A</sub> in VFY\_REQ shall be verified per 6.4.4 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of A,
2. The public verification key CP<sub>A</sub> which shall be retrieved from the Cert<sub>A</sub> in the payload of the ACT\_REQ,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for verification: ID<sub>S</sub> || ID<sub>R</sub> || NA' || NB || QA', and
5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

#### **9.2.3.3.2 Digital signature verification of Sig<sub>B</sub> in ACT\_RES**

The ECDSA digital signature Sig<sub>B</sub> in ACT\_RES shall be verified per 6.4.4 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of B,
2. The public verification key CP<sub>B</sub> which shall be retrieved from the Cert<sub>B</sub> in the payload of the ACT\_RES,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for verification: ID<sub>R</sub> || ID<sub>S</sub> || TTP' || NB' || NA || QB', and
5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

#### **9.2.3.3.3 Digital signature verification of Sig<sub>TTP</sub> in TAEP\_RES**

In the case of entity authentication involving a TTP, the ECDSA digital signature Sig<sub>TTP</sub> in TAEP\_RES shall be verified per 6.4.4 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of TTP,
2. The public verification key CS<sub>TTP</sub> which shall be retrieved from the Cert<sub>TTP</sub>,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for signing: NA || NB' || Res<sub>A</sub>' || Res<sub>B</sub>', and
5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

#### **9.2.3.3.4 Digital signature verification of Sig<sub>TTP</sub> in VFY\_REQ**

In the case of entity authentication involving a TTP, the ECDSA digital signature Sig<sub>TTP</sub> in VFY\_REQ shall be verified per 6.4.4 of ISO/IEC 14888-3, using:

1. Domain parameters according to the certificate of TTP,
2. The public verification key CS<sub>TTP</sub> which shall be retrieved from the Cert<sub>TTP</sub>,
3. A random value per ISO/IEC 18031, with the exception of Dual\_EC\_DRBG,
4. The message for signing: NA' || NB || Res<sub>A</sub>' || Res<sub>B</sub>', and

5. The Secure Hash Algorithm (SHA-256) per ISO/IEC 10118-3.

#### 9.2.4 Certificate validation

##### 9.2.4.1 TTP involved

###### 9.2.4.1.1 Certificate validation process

Certificate validation shall be performed by the TTP and include verification of the certificate's authenticity, expiration status and revocation status. Additional rules of validation may be required by the security policy of the security domain the TTP belongs to. For example, presence or absence of extensions, whether the subject name has the right format, whether the subject is granted access to the security domain, etc. The security policy and any additional rules are out of scope of this Standard.

During the certificate validation, if any check fails, then 'PDU content valid' shall be set to false.

###### 9.2.4.1.2 Res<sub>x</sub> generation

The certificate validation result of Sender (A) and Recipient (B), Res<sub>A</sub> and Res<sub>B</sub>, in TAEP\_RES shall be generated per 8.6.

###### 9.2.4.1.3 Res<sub>x</sub> verification

###### 9.2.4.1.3.1 Res<sub>x</sub> verification of Sender (A) in VFY\_REQ

The Recipient (B) shall check the certificate validation result of Sender (A), Res<sub>A</sub>', in VFY\_REQ as follows:

- a) check if the value of the Res Type subfield received from the Sender (A) in the Res<sub>A</sub>' of the payload of the VFY\_REQ equals to 0.
- b) check if the value of the Flag subfield received from the Sender (A) in the Res<sub>A</sub>' of the payload of the VFY\_REQ equals to 0.
- c) check if the Cert<sub>A</sub>' received from the Sender (A) in the ACT\_REQ is same as received in the Certificate subfield of Res<sub>A</sub>' of the payload of the VFY\_REQ.

###### 9.2.4.1.3.2 Res<sub>x</sub> verification of Recipient (B) in TAEP\_RES

The Sender (A) shall check the certificate validation result of Recipient (B), Res<sub>B</sub>', in TAEP\_RES as follows:

- a) check if the value of the Res Type subfield received from the TTP in the Res<sub>B</sub>' of the payload of the TAEP\_RES equals to 1.
- b) check if the value of the Flag subfield received from the TTP in the Res<sub>B</sub>' of the payload of the TAEP\_RES equals to 0.
- c) check if the Cert<sub>B</sub>' sent to the TTP in the payload of the TAEP\_REQ is the same as received in the Certificate subfield of Res<sub>B</sub>' of the payload of the TAEP\_RES.

##### 9.2.4.2 TTP not involved

Certificate validation shall include verification of the certificate's authenticity and expiration status, see ISO/IEC 9594-8. 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.

### 9.3 Key agreement

The ECDH shall be used for key agreement. Peer NFC-SEC entities shall agree on a shared secret using Key agreement mechanism 4 from ISO/IEC 11770-3 and the ECDH primitives including EC Key Pair Generation

primitive, EC Public key validation and ECDH secret value derivation Primitive per 9.1 of ISO/IEC 13157-3 (ECMA-409).

## 9.4 Key confirmation

This key confirmation mechanism is according to Clause 9 of ISO/IEC 11770-3. Key confirmation tag generation and verification per 11.4 of ISO/IEC 13157-2 (ECMA-386) shall be used after the key agreement mechanism.

## 9.5 Key Derivation Function (KDF)

After successful NEAU-A completion, the KDF per 9.2 of ISO/IEC 13157-3 (ECMA-409) shall be used to generate the shared secret Z and keys of SSE and SCH.

# 10 NEAU-A mechanism

## 10.1 Entity authentication involving a TTP

### 10.1.1 Protocol overview

NEAU-A mechanism involving a TTP is illustrated in Figure 6. During the NEAU-A, if any check fails, then 'PDU content valid' shall be set to false.

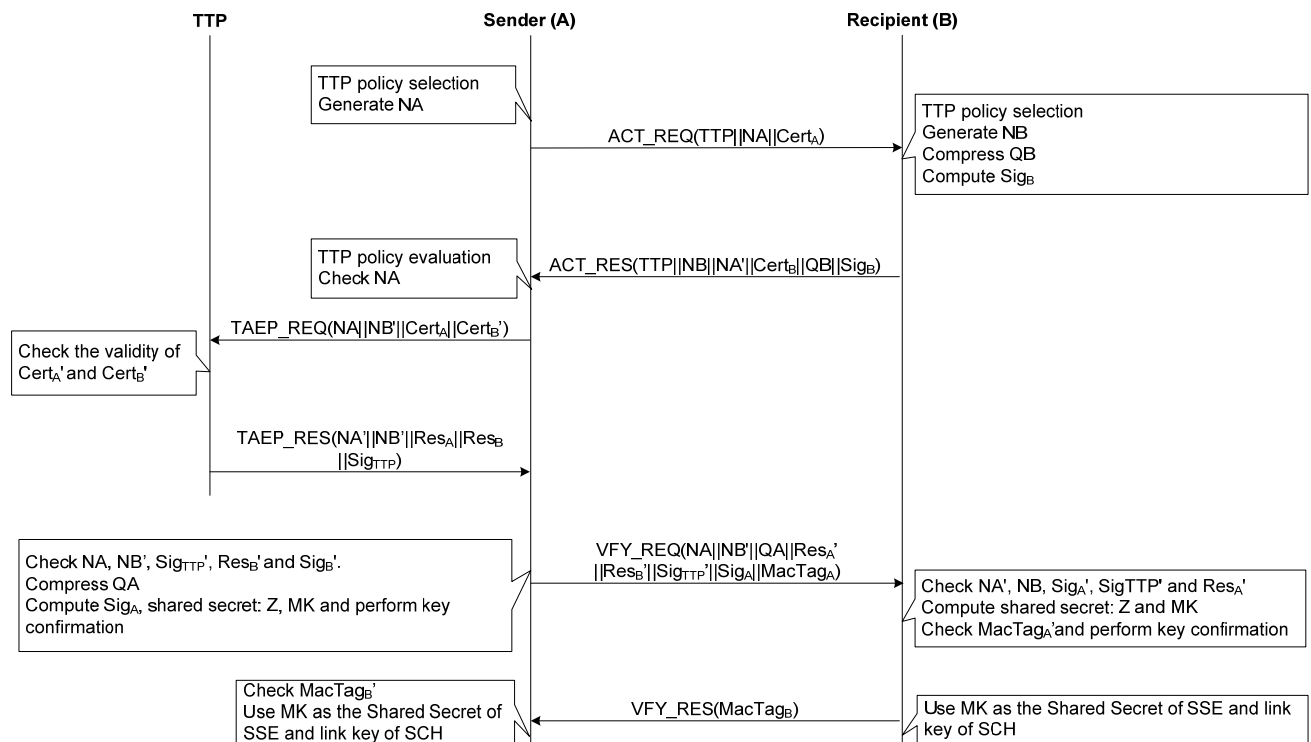


Figure 6 — NEAU-A mechanism involving a TTP overview

### 10.1.2 Preparation

Before starting the NEAU-A mechanism, the following shall be available to each NFC-SEC entity:

- Its own certificate conforming to ISO/IEC 9594-8/ITU X.509 and its own private key.
- TTP's certificate conforming to ISO/IEC 9594-8/ITU X.509.

- Each NFC-SEC entity shall be in possession of the n-entity-title of both peer-n-entities of the underlying n-layer.
- Each NFC-SEC entity knows the value of its TTP Policy<sub>x</sub>.
- The Sender (A) shall know if the TTP is available.

NOTE The TTP shall have a capability to validate the Sender's and Receiver's certificates according to the security policy of the security domain the TTP belongs to. The specification of the security policy and the TTP's validation capability is out of scope of this Standard.

NOTE The NFCIP-1-entity-title is the nfcid3 per ISO/IEC 18092, the 14443-3-entity-title is the UID.

### 10.1.3 Sender (A) transformation

1. Set the value of TTP field per 8.3.2.
2. Generate a nonce NA per 9.1.5 of ISO/IEC 13157-3 (ECMA-409).
3. Send TTP || NA || Cert<sub>A</sub> as the payload of the ACT\_REQ.
4. Receive TTP' || NB' || NA || Cert<sub>B</sub>' || QB' || Sig<sub>B</sub>' from the payload of the ACT\_RES.
5. Evaluate the value of TTP' field per 8.3.2.
6. If the verification of step a) is 'invalid', set the 'PDU content valid' to false in the protocol machine:
  - a) check if the random number NA sent to the Recipient (B) in the payload of the ACT\_REQ is the same as received in the ACT\_RES.
7. Send NA || NB' || Cert<sub>A</sub> || Cert<sub>B</sub>' as the payload of the TAEP\_REQ to the TTP according to ISO/IEC 9798-3:1998/Amd.1.
8. Receive NA || NB' || Res<sub>A</sub>' || Res<sub>B</sub>' || Sig<sub>TTP</sub>' as the payload of the TAEP\_RES from TTP.
9. Perform the following:
  - a) check if the random number NA send to the TTP in the payload of the TAEP\_REQ is the same as received in the TAEP\_RES;
  - b) check if the random number NB' send to the TTP in the payload of the TAEP\_REQ are the same as received in the TAEP\_RES and ACT\_RES, respectively;
  - c) check Sig<sub>TTP</sub>' per 9.2.3.3.3;
  - d) check Res<sub>B</sub>' per 9.2.4.1.3.2;
  - e) check Sig<sub>B</sub>' per 9.2.3.3.2.
10. Ensure QA equals the octet string of Q<sub>A</sub> per 10.3 of ISO/IEC 13157-2 (ECMA-386).
11. Compute Sig<sub>A</sub> per 9.2.3.2.1.

12. Reconstruct  $Q_B'$  from  $Q_B$  per 10.4 of ISO/IEC 13157-2 (ECMA-386). If the public EC keys have already been received, the previously calculated and stored value  $Q_B'$  may be reused and step 13, 14 and 15 may be skipped.
13. Verify that  $Q_B'$  is a valid key for the EC parameters per 9.1.3 of ISO/IEC 13157-3 (ECMA-409).
14. Use the Diffie-Hellman primitive in 9.1.4 of ISO/IEC 13157-3 (ECMA-409). If its output  $z$  is 'invalid', set the 'PDU content valid' to false in the protocol machine.
15. Convert  $z$  to octet string  $Z$  using the convention per 10.1 of ISO/IEC 13157-2 (ECMA-386).
16. Compute the MK and  $MacTag_A$  per 9.2 and 9.4.1 of ISO/IEC 13157-3 (ECMA-409) for key confirmation, respectively.
17. Send  $NA \parallel NB' \parallel QA \parallel Res_A' \parallel Res_B' \parallel Sig_{TTP}' \parallel Sig_A \parallel MacTag_A$  as payload of the VFY\_REQ.
18. Receive  $MacTag_B'$  from the payload of the VFY\_RES.
19. Check the key confirmation tag received from Recipient (B):  $MacTag_B'(MK)$  per 11.4.1 of ISO/IEC 13157-2 (ECMA-386).
20. Set the 'PDU content valid' to true, use MK as the Shared Secret of SSE and link key of SCH respectively.

#### 10.1.4 Recipient (B) transformation

1. Receive  $TTP' \parallel NA' \parallel Cert_A'$  from the payload of the ACT\_REQ.
2. Evaluate the value of the TTP' field and set the value of the TTP field per 8.3.2.
3. Generate a nonce NB per 9.1.5 of ISO/IEC 13157-3 (ECMA-409).
4. Ensure  $Q_B$  equals the octet string of  $Q_B$  per 10.3 of ISO/IEC 13157-2 (ECMA-386).
5. Compute  $Sig_B$  per 9.2.3.2.2.
6. Send  $TTP \parallel NB \parallel NA' \parallel Cert_B \parallel Q_B \parallel Sig_B$  as the payload of the ACT\_RES.
7. Receive  $NA' \parallel NB \parallel QA \parallel Res_A \parallel Res_B \parallel Sig_{TTP} \parallel Sig_A$  from the payload of the VFY\_REQ.
8. Perform the following:
  - a) check if the random number  $NA'$  received from the Sender (A) in the payload of the ACT\_REQ is the same as received in the VFY\_REQ;
  - b) check if the random number NB sent to the Sender (A) in the payload of the ACT\_RES is the same as received in the VFY\_REQ;
  - c) check  $Sig_A'$  per 9.2.3.3.1;
  - d) check  $Sig_{TTP}'$  per 9.2.3.3.4;
  - e) check  $Res_A'$  per 9.2.4.1.3.1.

9. Reconstruct  $Q_A'$  from  $QA'$  per 10.4 of ISO/IEC 13157-2 (ECMA-386). If the public EC keys have already been received, the previously calculated and stored value  $Q_A'$  may be reused and step 10, 11 and 12 may be skipped.
10. Verify that  $QA'$  is a valid key for the EC parameters per 9.1.3 of ISO/IEC 13157-3 (ECMA-409).
11. Use the Diffie-Hellman primitive in 9.1.4 of ISO/IEC 13157-3 (ECMA-409). If its output  $z$  is 'invalid', set the 'PDU content valid' to false in the protocol machine.
12. Convert  $z$  to octet string  $Z$  using the convention per 10.1 of ISO/IEC 13157-2 (ECMA-386).
13. Compute the MK per 9.2 of ISO/IEC 13157-3 (ECMA-409).
14. Check the key confirmation tag received from Sender (A):  $MacTag_A'(MK)$  per 11.4.2 of ISO/IEC 13157-2 (ECMA-386).
15. Compute  $MacTag_B$  per 9.4.1 of ISO/IEC 13157-3 (ECMA-409) and send it as the payload of the VFY\_REQ.
16. Set the 'PDU content valid' to true, use MK as the Shared Secret of SSE and link key of SCH respectively.

#### 10.1.5 TTP transformation

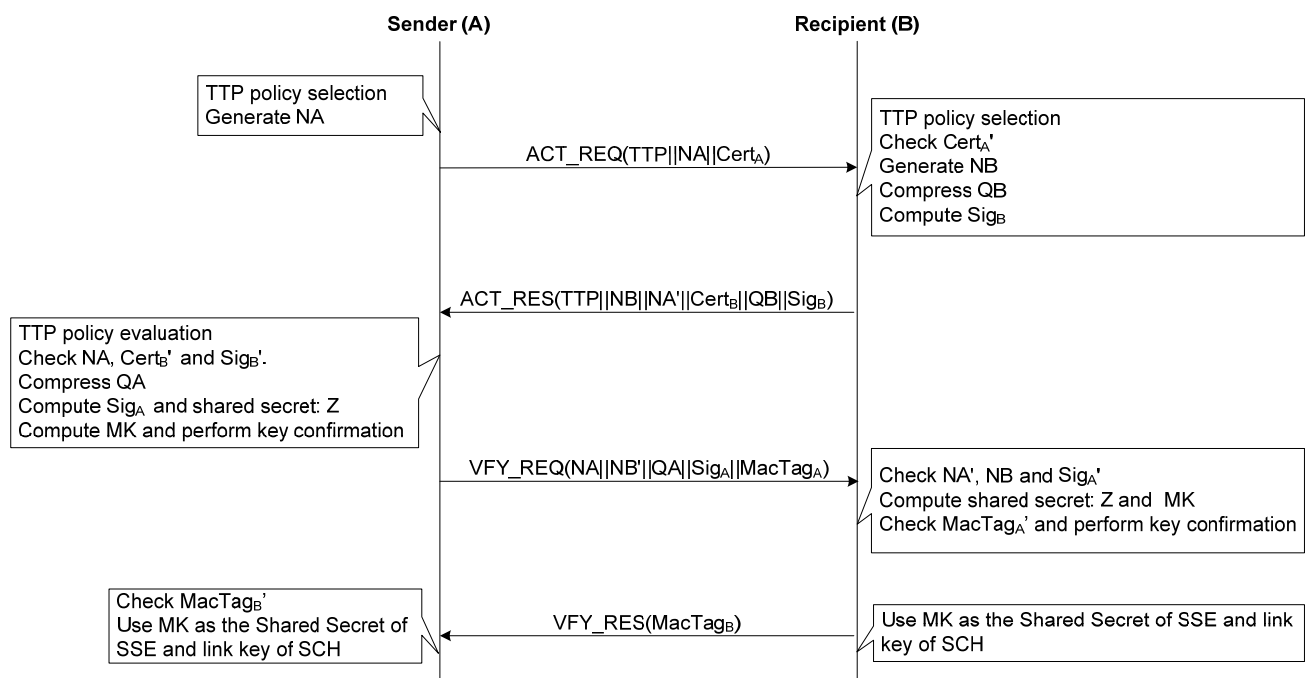
If the field of TTP is set to 1 and the TTP is not involved, TTP transformation shall be skipped. However, the capability of involving a TTP is mandatory in this Standard. The PDUs exchanges between TTP and the Sender (A) are transmitted by the other channel and based on the UDP segment with Port 5111 and TAEP encapsulation. See Annex A for more details about UDP Port 5111 and TAEP.

1. Receive  $NA' || NB' || Cert_A' || Cert_B'$  from the payload of the TAEP\_REQ.
2. Validate  $Cert_A'$  and  $Cert_B'$  per 9.2.4.1.1.
3. Generate  $Res_A$  and  $Res_B$  per 9.2.4.1.2.
4. Compute  $Sig_{TTP}$  per 9.2.3.2.3.
5. Send  $NA' || NB' || Res_A || Res_B || Sig_{TTP}$  as the payload of the TAEP\_RES.

## 10.2 Entity authentication without involving a TTP

### 10.2.1 Protocol overview

NEAU-A mechanism without involving a TTP is illustrated in Figure 7. During the NEAU-A, if any check fails, then 'PDU content valid' shall be set to false.



**Figure 7 — NEAU-A mechanism without involving a TTP overview**

### 10.2.2 Preparation

Before starting the NEAU-A mechanism, the following shall be available on each NFC-SEC entity:

- Its own certificate conforming to ISO/IEC 9594-8/ITU X.509 and its own private key.
- The capability to validate the peer's certificate.
- Each NFC-SEC entity shall be in possession of the n-entity-title of both peer-n-entities of the underlying n-layer.
- Each NFC-SEC entity knows the value of its TTP Policy<sub>x</sub>.
- The Sender (A) shall know if the TTP is available.

### 10.2.3 Sender (A) transformation

1. Set the value of TTP field per 8.3.2.
2. Generate a nonce NA per 9.1.5 of ISO/IEC 13157-3 (ECMA-409).
3. Send TTP || NA || Cert<sub>A</sub> as the payload of the ACT\_REQ.
4. Receive TTP' || NB' || NA || Cert<sub>B</sub>' || QB' || Sig<sub>B</sub>' from the payload of the ACT\_RES.
5. Evaluate the value of TTP' field per 8.3.2.
6. Perform the following:
  - a) check if the random number NA sent to the Recipient (B) in the payload of the ACT\_REQ is the same as received in the ACT\_RES;

- b) Check Cert<sub>B</sub>' per 9.2.4.2;
  - c) Check Sig<sub>B</sub>' per 9.2.3.3.2.
7. Ensure QA equals the octet string of Q<sub>A</sub> per 10.3 of ISO/IEC 13157-2 (ECMA-386).
  8. Compute Sig<sub>A</sub> per 9.2.3.2.1.
  9. Reconstruct QB' from QB' per 10.4 of ISO/IEC 13157-2 (ECMA-386). If the public EC keys have already been received, the previously calculated and stored value QB' may be reused and step 10, 11 and 12 may be skipped.
  10. Verify that QB' is a valid key for the EC parameters per 9.1.3 of ISO/IEC 13157-3 (ECMA-409).
  11. Use the Diffie-Hellman primitive in 9.1.4 of ISO/IEC 13157-3 (ECMA-409). If its output z is 'invalid', set the 'PDU content valid' to false in the protocol machine.
  12. Convert z to octet string Z using the convention per 10.1 of ISO/IEC 13157-2 (ECMA-386).
  13. Compute the MK and MacTag<sub>A</sub> per 9.2 and 9.4.1 of ISO/IEC 13157-3 (ECMA-409) for key confirmation, respectively.
  14. Send NA || NB' || QA || Sig<sub>A</sub> || MacTag<sub>A</sub> as the payload of the VFY\_REQ.
  15. Receive MacTag<sub>B</sub>' from the payload of the VFY\_RES.
  16. Check the key confirmation tag received from Recipient (B): MacTag<sub>B</sub>'(MK) per 11.4.1 of ISO/IEC 13157-2 (ECMA-386).
  17. Set the 'PDU content valid' to true, use MK as the Shared Secret of SSE and link key of SCH respectively.

#### 10.2.4 Recipient (B) transformation

1. Receive TTP' || NA' || Cert<sub>A</sub>' from the payload of the ACT\_REQ.
2. Check Cert<sub>A</sub>' per 9.2.4.2.
3. Evaluate the value of TTP' field and set the TTP field per 8.3.2.
4. Generate a nonce NB per 9.1.5 of ISO/IEC 13157-3 (ECMA-409).
5. Ensure QB equals the octet string of Q<sub>B</sub> per 10.3 of ISO/IEC 13157-2 (ECMA-386).
6. Compute Sig<sub>B</sub> per 9.2.3.2.2.
7. Send TTP || NB || NA' || Cert<sub>B</sub> || QB || Sig<sub>B</sub> as the payload of the ACT\_RES.
8. Receive NA' || NB || QA' || Sig<sub>A</sub>' || MacTag<sub>A</sub>' from the payload of the VFY\_REQ.
9. Perform the following:
  - a) check if the random number NA' received from the Sender (A) in the payload of the ACT\_REQ is the same as received in the VFY\_REQ;



- b) check if the random number NB sent to the Sender (A) in the payload of the ACT\_RES is the same as received in the VFY\_REQ;
  - c) check  $\text{Sig}_A'$  per 9.2.3.3.1.
10. Reconstruct  $Q_A'$  from  $Q_A$  per 10.4 of ISO/IEC 13157-2 (ECMA-386). If the public EC keys have already been received, the previously calculated and stored value  $Q_A'$  may be reused and step 11, 12 and 13 may be skipped.
  11. Verify that  $Q_A'$  is a valid key for the EC parameters per 9.1.3 of ISO/IEC 13157-3 (ECMA-409).
  12. Use the Diffie-Hellman primitive in 9.1.4 of ISO/IEC 13157-3 (ECMA-409). If its output  $z$  is 'invalid', set the 'PDU content valid' to false in the protocol machine.
  13. Convert  $z$  to octet string  $Z$  using the convention per 10.1 of ISO/IEC 13157-2 (ECMA-386).
  14. Compute the MK per 9.2 of ISO/IEC 13157-3 (ECMA-409).
  15. Check the key confirmation tag received from Sender (A):  $\text{MacTag}_A'(\text{MK})$  per 11.4.2 of ISO/IEC 13157-2 (ECMA-386).
  16. Compute  $\text{MacTag}_B$  per 9.4.1 of ISO/IEC 13157-3 (ECMA-409) and send it as the payload of the VFY\_REQ.
  17. Set the 'PDU content valid' to true, use MK as the Shared Secret of SSE and link key of SCH respectively.

### 10.3 Key derivation

#### 10.3.1 Sender (A)

Sender (A)'s private EC key  $d_A$  shall be obtained from a random or pseudo-random generation process conforming to ISO/IEC 18031. Sender (A)'s public EC key  $Q_A = d_A \cdot G$ .

Based on the ECDH protocol,  $z = d_A \cdot Q_B = d_B \cdot Q_A$ .

Convert  $z$  to octet string  $Z$  per 10.1 of ISO/IEC 13157-2 (ECMA-386).

For the SSE service, derive  $\text{MK}_{\text{SSE}} = \text{KDF-SSE}(\text{NA}, \text{NB}', \text{ID}_S, \text{ID}_R, Z)$  per 9.2.1 of ISO/IEC 13157-3 (ECMA-409).

For the SCH service, derive  $\{\text{MK}_{\text{SCH}}, \text{K}_{\text{SCH}}\} = \text{KDF-SCH}(\text{NA}, \text{NB}', \text{ID}_S, \text{ID}_R, Z)$  per 9.2.2 of ISO/IEC 13157-3 (ECMA-409).

#### 10.3.2 Recipient (B)

Recipient (B)'s private EC key  $d_B$  shall be obtained from a random or pseudo-random generation process conforming to ISO/IEC 18031. Recipient (B)'s public EC key  $Q_B = d_B \cdot G$ .

Based on the ECDH protocol,  $z = d_B \cdot Q_A = d_A \cdot Q_B$ .

Convert  $z$  to octet string  $Z$  per 10.1 of ISO/IEC 13157-2 (ECMA-386).

For the SSE service, derive  $\text{MK}_{\text{SSE}} = \text{KDF-SSE}(\text{NA}', \text{NB}, \text{ID}_S, \text{ID}_R, Z)$  per 9.2.1 of ISO/IEC 13157-3 (ECMA-409).

For the SCH service, derive  $\{MK_{SCH}, K_{SCH}\} = \text{KDF-SCH}(NA', NB, ID_s, ID_R, Z)$  per 9.2.2 of ISO/IEC 13157-3 (ECMA-409).

## **11 Data Authenticated Encryption in SCH**

Clause 12 of ISO/IEC 13157-3 (ECMA-409) applies.

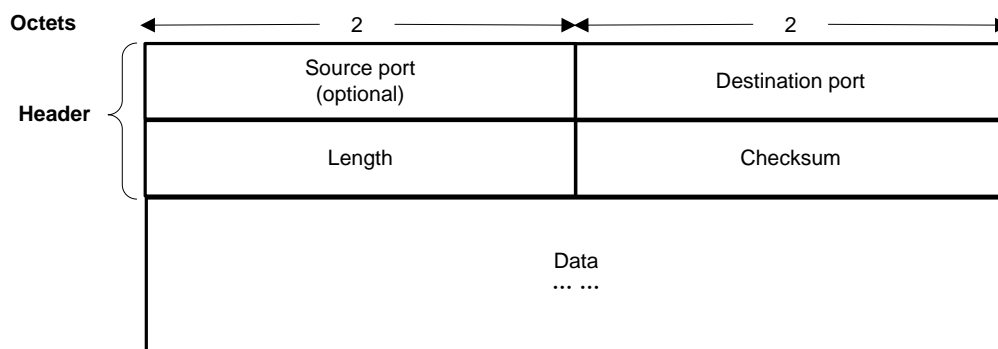
## Annex A (normative)

### UDP Port 5111 and TAEP

#### A.1 UDP and port 5111

##### A.1.1 UDP

The User Datagram Protocol (UDP) is one of the core members of the Internet protocol suite. With UDP, computer applications can send messages, in this case referred to as datagrams, to other hosts on an Internet Protocol (IP) network without prior communications to set up special transmission channels or data paths. UDP is a minimal message-oriented Transport Layer protocol. Figure A.1 shows the UDP segment format. See RFC 768 for more details.



**Figure A.1 — UDP segment format**

1. Source port: See RFC 768 for the definition of Source port field and Annex A.1.2 for more details of the number of Source port in NEAU-A mechanism.
2. Destination port: See RFC 768 for the definition of Destination port field and Annex A.1.2 for more details of the number of Destination port in NEAU-A mechanism.
3. Length: See RFC 768 for the definition of Length field.
4. Checksum: See RFC 768 for the definition of Checksum field.
5. Data: See RFC 768 for the definition of Data field. TEAP packets shall be encapsulated in this field in NEAU-A mechanism (See Annex A.2).

##### A.1.2 Port 5111

In this Standard, the PDU exchange between TTP and the Sender (A) is based on the UDP segment with Port 5111.

Sender (A) shall send the UDP segment with an ephemeral Source port number and 5111 Destination port number.

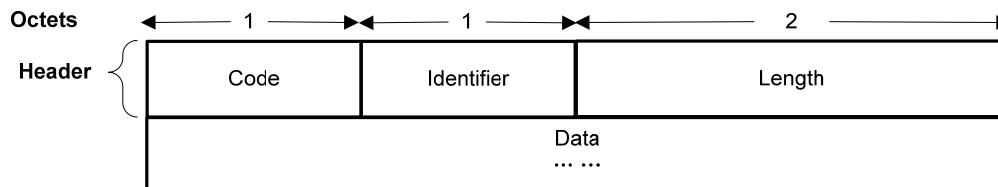
TTP shall answer sender (A) by send the UDP segment with 5111 Source port number. The value of Destination port number is the same as the value of Source port number send by Sender (A).

NOTE The UDP port 5111 has been assigned by Internet Assigned Numbers Authority for TAEP authentication service.

## A.2 TAEP

### A.2.1 TAEP packet format

Tri-element Authentication Extensible Protocol (TAEP) is worked with UDP port 5111 to provide the entity authentication service. TEAP packets shall be encapsulated in the Date field of PDU segment in NEAU-A mechanism Figure A.2 specifies the TAEP packet format.



**Figure A.2 — TAEP packet format**

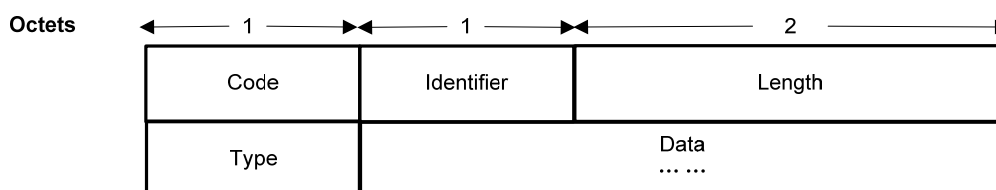
1. Code: The Code field shall be 1 octet and identifies the Type of TEAP packet. TEAP Codes are assigned as follows:
  - a) Request (TAEP\_REQ)
  - b) Response (TAEP\_RES)
  - c) In this Standard, the Code values 3-255 are reserved.
2. Identifier: The Identifier field shall be 1 octet and aids in matching Responses with Requests. In this Standard, the Identifier field shall be changed on each TAEP\_REQ. TAEP\_RES shall match the Identifier field from the corresponding TAEP\_REQ.
3. Length: The Length field shall be 2 octets and specifies the length in octets of the TAEP header and TAEP data. The minimum length is 4 octets since that's the length of the header. Octets outside the range of the Length field shall be ignored upon reception. A message with the Length field set to a value larger than the number of received octets shall be silently discarded.
4. Data: The Data field is zero or more octets and consists of the encapsulated message per A.2.2.

### A.2.2 TAEP\_REQ and TAEP\_RES format

A summary of the TAEP\_REQ and TAEP\_RES formats is specified in Figure A.3.

In this Standard,

1. The Type value in TAEP\_REQ and TAEP\_RES shall be set to 5.
2. Sender (A) shall use TAEP\_REQ to encapsulate message  $NA \parallel NB' \parallel Cert_A \parallel Cert_B'$  into the Data field.
3. TTP shall use TAEP\_RES to encapsulate message  $NA' \parallel NB' \parallel Res_A \parallel Res_B \parallel Sig_{TTP}$  into the Data field.



**Figure A.3 —TAEP\_REQ and TAEP\_RES format**

## Annex B (informative)

### ECDSA test vectors

The test vectors in the annex may be used to verify implementations of ECDSA using curve P-256.

Key pair: (qlen = 256 bits)

q = FFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551

private key:

d = 17D5BE4BC2368948C5B376F38A12B5E2D266B951E034A325E394E2D0BAB3628D

public key: a point on EC, PK = d·G

PK<sub>x</sub> = F77E59501B5CE6A85783702E79EF3EA38E415E5F3E5E09EFCFF9117E8457711C

PK<sub>y</sub> = 16431599C76D48D49A933123997EA06D95E5F334158A06C894D6FAF2181E7367

Signatures:

With SHA-256, message = "NFC-SEC":

k = DEC8F06306B78F6E1784EF851687E3935049A39F90AFB5A7ACF9363BE185FA82

Signature (r, s):

r = 7E20D307AF54DFBED17E37A064B72FF5E6A2B832F93EF37B2BB2D8B8124EC143

s = F53777F606FB448228702EBB0AC584B96684AD13D6412CCAD052747C637539CD

With SHA-256, message = "NEAU":

k = CDB85E97D49902C66FFAB0A1200A01A3A1A155C3B2F565B69EE5B08B470B91D5

Signature (r, s):

r = DE6BE243414FD55053C4634179757FBBC2D01313B4FEA163E5C5C18AA0FAFAA2

s = F0E1B19EC105C9DA38A439A0E0A3CEAE1210184F21FD985CC401AEC17F37B127

## Bibliography

- [1] Internet Assigned Numbers Authority  
<http://www.iana.org/>
- [2] RFC 768, *User Datagram Protocol*
- [3] RFC 6979, *Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)*
- [4] RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*.



