

**3<sup>rd</sup> Generation Partnership Project;  
Technical Specification Group Services and System Aspects;  
5G Security Assurance Specification (SCAS) for  
the Security Edge Protection Proxy (SEPP)  
network product class  
(Release 19)**



---

Keywords  
SCAS,5G,SEPP,security

**3GPP**

---

Postal address

---

3GPP support office address  
650 Route des Lucioles – Sophia Antipolis  
Valbonne – FRANCE  
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

---

Internet

---

<http://www.3gpp.org>

---

# Contents

Foreword.....	4
1 Scope.....	6
2 References.....	6
3 Definitions of terms, symbols and abbreviations.....	6
3.1 Terms.....	6
3.2 Symbols.....	6
3.3 Abbreviations.....	6
4 SEPP-specific security requirements and related test cases.....	7
4.1 Introduction.....	7
4.2 SEPP-specific adaptations of security functional requirements and related test cases.....	7
4.2.1 Introduction.....	7
4.2.2 Security functional requirements on the SEPP deriving from 3GPP specifications and related test cases.....	7
4.2.2.1 Security functional requirements on the SEPP deriving from 3GPP specifications – general approach.....	7
4.2.2.2 Correct handling of cryptographic material of peer SEPPs and IPX providers.....	7
4.2.2.3 Connection-specific scope of cryptographic material by IPX-providers.....	9
4.2.2.4 Correct handling of serving PLMN ID mismatch.....	10
4.2.2.5 Confidential IEs replacement handling in original N32-f message.....	11
4.2.2.6 Correct handling of protection policy mismatch.....	11
4.2.2.7 JWS profile restriction.....	13
4.2.2.8 No misplacement of encrypted IEs in JSON object by IPX.....	14
4.2.2.9 Correct Handling of Inter-PLMN Routing.....	15
4.2.2.10 Correct Handling of Custom HTTP Header with PRINS Security.....	16
4.2.3 Technical Baseline.....	17
4.2.3.1 Introduction.....	17
4.2.3.2 Protecting data and information.....	17
4.2.3.2.1 Protecting data and information – general.....	17
4.2.3.2.2 Protecting data and information – unauthorized viewing.....	17
4.2.3.2.3 Protecting data and information in storage.....	17
4.2.3.2.4 Protecting data and information in transfer.....	17
4.2.3.2.5 Logging access to personal data.....	17
4.2.3.3 Protecting availability and integrity.....	17
4.2.3.4 Authentication and authorization.....	17
4.2.3.5 Protecting sessions.....	17
4.2.3.6 Logging.....	18
4.2.4 Operating Systems.....	18
4.2.5 Web Servers.....	18
4.2.6 Network Devices.....	19
4.3 SEPP-specific adaptations of hardening requirements and related test cases.....	19

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2025, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  
All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
GSM® and the GSM logo are registered and owned by the GSM Association

4.3.1	Introduction.....	19
4.3.2	Technical baseline.....	19
4.3.3	Operating systems.....	19
4.3.4	Web servers.....	20
4.3.5	Network devices.....	20
4.3.6	Network functions in service-based architecture.....	20
4.4	SEPP-specific adaptations of basic vulnerability testing requirements and related test cases.....	20
4.4.1	Introduction.....	20
4.4.2	Port Scanning.....	20
4.4.3	Vulnerability scanning.....	20
4.4.4	Robustness and fuzz testing.....	20
	Annex A (informative): Change history.....	21

---

## Foreword

This Technical Specification has been produced by the 3<sup>rd</sup> Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

- 1 presented to TSG for information;
- 2 presented to TSG for approval;
- 3 or greater indicates TSG approved document under change control.

Y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

---

## 1 Scope

The present document contains objectives, requirements and test cases that are specific to the SEPP network product class. It refers to the Catalogue of General Security Assurance Requirements and formulates specific adaptions of the requirements and test cases given there, as well as specifying requirements and test cases unique to the SEPP network product class.

---

## 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [2] 3GPP TS 33.117: "Catalogue of General Security Assurance Requirements".
- [3] 3GPP TS 33.501: "Security architecture and procedures for 5G system".
- [4] 3GPP TR 33.926: "Security Assurance Specification (SCAS) threats and critical assets in 3GPP network product classes".
- [5] Void.
- [6] 3GPP TS 29.573: "5G System; Public Land Mobile Network (PLMN) Interconnection".
- [7] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture".
- [8] 3GPP TS 23.501: "System Architecture for 5G System (5GS)".

---

## 3 Definitions of terms, symbols and abbreviations

### 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

### 3.2 Symbols

Void.

### 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

IPX	IP eXchange service
JSON	JavaScript Object Notation
JWS	JSON Web Signature
NF	Network Function
SEPP	Security Edge Protection Proxy

---

## 4 SEPP-specific security requirements and related test cases

### 4.1 Introduction

SEPP specific security requirements include both requirements derived from SEPP-specific security functional requirements in relevant specifications as well as security requirements introduced in the present document derived from the threats specific to SEPP as described in TR 33.926 [4].

### 4.2 SEPP-specific adaptations of security functional requirements and related test cases

#### 4.2.1 Introduction

The present clause describes the security functional requirements and the corresponding test cases for SEPP network product class. The proposed security requirements are classified in two groups:

- Security functional requirements derived from TS 33.501 [3] and detailed in clause 4.2.2.
- General security functional requirements which include requirements not already addressed in TS 33.501 [3] but whose support is also important to ensure that SEPP conforms to a common security baseline detailed in clause 4.2.3.

#### 4.2.2 Security functional requirements on the SEPP deriving from 3GPP specifications and related test cases

##### 4.2.2.1 Security functional requirements on the SEPP deriving from 3GPP specifications – general approach

In addition to the requirements and test cases in TS 33.117 [2], clause 4.2.2.1, a SEPP shall satisfy the following:

- It is assumed for the purpose of the present SCAS that a SEPP conforms to all mandatory security-related provisions pertaining to an SEPP in:
    - 3GPP TS 33.501 [3]: "Security architecture and procedures for 5G system";
    - other 3GPP specifications that make reference to TS 33.501 [3] or are referred to from TS 33.501 [3]
- Security procedures pertaining to a SEPP are typically embedded in NF/NF service status discovery/subscribe/notify procedures across PLMNs and are hence assumed to be tested together with them.

##### 4.2.2.2 Correct handling of cryptographic material of peer SEPPs and IPX providers

*Requirement Name:* Correct handling of cryptographic material of peer SEPPs and IPX providers

*Requirement Reference:* TS 33.501 [3], clause 5.9.3.2

*Requirement Description:*

The SEPP is able to clearly differentiate between certificates used for authentication of peer SEPPs and certificates used for authentication of intermediates performing message modifications.

*Threat References:* TR 33.926 [4], clause G.2.2.1, Misusing cryptographic material of peer SEPPs and IPX providers

*Test Case:*

**Test Name:** TC\_CRYPT\_MATERIAL\_SEPP\_IPX\_SEPARATION

*Purpose:*

Verify that the SEPP under test does not accept raw public keys/certificates by intermediate IPX-providers for N32-c TLS connection establishment. The opposite is to be ensured as well: The SEPP under test shall not accept N32-f JSON patches signed with raw public keys/certificates of peer SEPPs.

#### **Procedure and execution steps:**

##### **Pre-Conditions:**

- System documentation of the SEPP under test, which details how raw public keys/certificates of peer SEPPs are to be configured and how internal log files can be accessed.
- A second SEPP instance for N32 communication with the SEPP under test, which allows for the creation of custom N32-f messages. This system may be simulated.
- Both SEPPs are to be configured with a raw public key/certificate of their communication peer to be able to establish a N32-c connection.
- Test environment with one node simulating an IPX-provider. This functionality includes parsing N32-f messages, creation of JSON-patches for message modifications and JWS operations, among others.
- Two public/private key pairs representing IPX-providers. These cryptographic keys need to be different from those of the two SEPPs.

##### **Execution Steps**

- 1.1 Both SEPPs are configured for N32-f communication via the simulated IPX-system.
- 1.2 Both SEPPs establish a N32 connection with each other. The secondary SEPP provides the IPX-provider's public key/certificate to the SEPP under test as part of the *IPX security information list* via N32-c.
- 1.3 While the N32 connection from the previous step is still active, the tester attempts to establish an additional N32-c TLS connection using the IPX-providers private key.
- 1.4 Based on the internal log files, the tester validates how the SEPP under test handles the N32-c connection attempt.
- 2.1 Both SEPPs are configured for N32-f communication via the simulated IPX-system.
- 2.2 Both SEPPs establish a N32-c connection with each other. The secondary SEPP provides the IPX-provider's public key/certificate to the SEPP under test as part of the *IPX security information list* via N32-c.
- 2.3 The tester sends a N32-f message from the secondary SEPP via the IPX-system towards the SEPP under test.
- 2.4 The intermediate IPX-system appends an arbitrary JSON-(NULL-)patch to the N32-f message and signs it not with its own private key, but the private key of the secondary SEPP. The modified message is then forwarded to the SEPP under test.
- 2.5 Based on the internal log files, the tester validates how the received N32-f message is handled by the SEPP under test.

##### **Expected Results:**

- The N32-c TLS connection establishment using the cryptographic material of the intermediate IPX-system fails with the SEPP to be tested (step 1.4).
- The JSON patch signed with the peer SEPP's private key is discarded by the SEPP under test (step 2.5).

##### **Expected format of evidence:**

Logs and the communication flow saved in a .pcap file.

#### **4.2.2.3 Connection-specific scope of cryptographic material by IPX-providers**

*Requirement Name:* Connection-specific scope of cryptographic material by IPX-providers

*Requirement Reference:* TBA

*Requirement Description:*

Cryptographic material from IPX providers, i.e. raw public keys or certificates, used to authenticate N32-f message modifications is only valid for the N32 connection it is exchanged in. The SEPP under test shall not accept N32-f message modifications signed by IPX-providers other than the ones whose cryptographic material has been exchanged as part of the *IPX security information list* via the related N32-c connection.

*Threat References:* TR 33.926 [4], clause G.2.2.2, Misusing cryptographic material beyond connection-specific scope

*Test Case:*

**Test Name:** TC\_CONNECTION\_SPECIFIC\_SCOPE\_CRYPT\_MATERIAL

**Purpose:**

Verify that the SEPP to be tested does not use cryptographic material from IPX-providers other than the ones whose raw public key/certificate has been exchanged in the related N32-c connection to authenticate N32-f message modifications.

**Procedure and execution steps:**

**Pre-Conditions:**

- System documentation of the SEPP under test, which details how raw public keys/certificates of peer SEPPs are to be configured and how internal log files can be accessed.
- Test environment with one node simulating an IPX-provider. This functionality includes parsing N32-f messages, creation of JSON-patches for message modifications and JWS operations, among others.
- Two public/private key pairs representing IPX-providers.
- A second SEPP instance for N32 communication with the SEPP under test, which allows for the creation of custom N32-f messages. This system may be simulated.
- Both SEPPs are to be configured with the raw public key/certificate of their communication peer to be able to establish an N32-c TLS connection.

**Execution Steps**

1. Both SEPPs are configured for N32-f communication via the simulated IPX-system.
2. Both SEPPs establish a mutual N32-c connection. As part of the *IPX security information list*, the secondary SEPP provides one of the prepared raw public keys/certificates of the IPX-providers (KEY\_A) to the SEPP under test.
3. Parallel to the N32 connection in step 1, an additional connection is established between the two SEPPs. Within this connection, an alternate raw public key/certificate of the IPX-providers (KEY\_B) shall be exchanged.
4. Within the N32 connection established in step 1, the tester sends an N32-f message from the secondary SEPP towards the SEPP under test. The intermediate IPX-system appends an arbitrary JSON-(NULL-)patch, which is signed with the private key belonging to KEY\_B, i.e. the one out of scope of this particular N32 connection. The modified message is then forwarded to the SEPP to be tested.
5. Based on the log files of the SEPP under test, the tester validates how the received N32-f message is handled.

**Expected Results:**

- N32-f message modifications which have been signed by IPX-providers whose information has not been exchanged as part of the related N32-c connection are discarded by the SEPP under test.

**Expected format of evidence:**

Logs and the communication flow saved in a .pcap file.

#### 4.2.2.4      Correct handling of serving PLMN ID mismatch

*Requirement Name:* Correct handling of serving PLMN ID mismatch

*Requirement Reference:* TS 33.501 [3], clause 13.2.4.7, and TS 33.501 [3], clause 13.4.1.2

*Requirement Description:*

The receiving SEPP verifies that the PLMN-ID contained in the incoming N32-f message matches the PLMN-ID in the related N32-f context as specified in TS 33.501 [3], clause 13.2.4.7.

The pSEPP checks that the serving PLMN ID of subject claim in the access token matches the remote PLMN ID corresponding to the N32-f context Id in the N32 message" as specified in TS 33.501 [3], clause 13.4.1.2.

*Threat References:* TR 33.926 [4], clause G.2.3.1, Incorrect handling for PLMN ID mismatch

*Test case:*

**Test Name:** TC\_PLMN\_ID\_MISMATCH

**Purpose:**

Verify that the SEPP under test is able to identify the mismatch between the PLMN-ID contained in the incoming N32-f message and the PLMN-ID in the related N32-f context, and take action accordingly.

**Procedure and execution steps:**

**Pre-Conditions:**

- Test environment with a peer SEPP instance (as cSEPP), which may be simulated.
- The SEPP under test and the peer SEPP have mutually authenticated and already established N32-c connection.
- The SEPP under test has established N32-f context with the peer SEPP. The SEPP under test is in possession of the N32-f peer information which contains remote PLMN ID of the peer SEPP.
- The tester shall have access to the interfaces of the SEPP under test and the peer SEPP.

**Execution Steps:**

1. The tester computes an access token correctly, except that the PLMN ID appended in the subject claim of the access token is different from PLMN ID of the peer SEPP, and then includes the access token in a NF Service Request.
2. The peer SEPP sends to the SEPP under test a N32 message containing the NF Service Request with the access token.
3. The SEPP under test receives the incoming N32 message from the peer SEPP and verifies that the PLMN ID in the subject claim of the access token does not match the remote PLMN ID in the N32-f peer information in the N32-f context.

**Expected Results:**

- The SEPP under test sends an error signalling message containing the N32-f Message Id and error code to the peer SEPP on the N32-c connection.

**Expected format of evidence:**

Logs and the communication flow saved in a .pcap file.

#### 4.2.2.5 Confidential IEs replacement handling in original N32-f message

*Requirement Name:* Confidential IEs replacement handling in original N32-f message

*Requirement Reference:* TS 29.573 [6], clause 5.3.2.3

*Requirement Description:*

Based on the protection policy exchanged between the SEPPs, the sending SEPP prepares an input for the JWE ciphering and integrity protection as an array of free form JSON objects in the "DataToIntegrityProtectAndCipher" block with each entry containing either a HTTP header value or the value of a JSON payload IE of the API message

being reformatted. The index value "encBlockIdx" in the payload part of DataToIntegrityProtectBlock points to the index of a header value or IE value in this input array.

*Threat References:* TR 33.926 [4], clause G.2.4.2, Exposure of confidential IEs in N32-f message

**Test Case:**

**Test Name:** TC\_SEPP\_CONFIDENTIAL\_IE\_REPLACEMENT\_N32F

**Purpose:**

Verify that the SEPP under test correctly replaces information elements requiring encryption with the value "encBlockIdx".

**Procedure and execution steps:**

**Pre-Conditions:**

- System documentation of the SEPP under test, which details how raw public keys/certificates of peer SEPPs are to be configured and how internal log files can be accessed.
- A second SEPP instance for N32 communication with the SEPP under test, which allows for the creation of custom N32-f messages. This system may be simulated.
- Both SEPPs are to be configured with a raw public key/certificate of their communication peer to be able to establish a N32-c connection.
- An arbitrary Data-type encryption policy which includes at least one information element requiring encryption on N32-f. The SEPP under test is to be configured with this policy.

**Execution Steps**

1. Both SEPPs establish a mutual N32-c connection.
2. Via the PLMN-internal interface, the tester provides the SEPP under test with a message to be forwarded to the peer SEPP on N32. This message needs to contain at least one information element that requires encryption according to the locally configured Data-type encryption policy.
3. The tester captures the related N32-f message after transformation by the SEPP under test.

**Expected Results:**

Information elements in the original message that require encryption according to the Data-type encryption policy are replaced with the value "encBlockIdx".

**Expected format of evidence:**

Evidence suitable for the interface, e.g. text representation of the captured N32-f message.

#### 4.2.2.6      Correct handling of protection policy mismatch

*Requirement Name:* Correct handling of protection policy mismatch

*Requirement Reference:* TS 33.501 [3], clause 13.2.3.6

*Requirement Description:*

"When a SEPP receives a data-type encryption or modification policy on N32-c as specified in clause 13.2.2.2, it compares it to the one that has been manually configured for this specific roaming partner and IPX provider. If a mismatch occurs for one of the two policies, the SEPP performs one of the following actions, according to operator policy:

- Send the error message as specified in TS 29.573 [6], clause 6.1.4.3.2, to the peer SEPP;
- Create a local warning".

*Threat References:* TR 33.926 [4], clause G.2.3.2, Incorrect handling for protection policy mismatch

*Test case:*

**Test Name:** TC\_SEPP\_POLICY\_MISMATCH

**Purpose:**

Verify that the SEPP under test is able to identify the mismatch between the protection policies manually configured for a specific roaming partner and IPX provider and the protection policies received on N32-c connection, and take action accordingly.

**Procedure and execution steps:**

**Pre-Conditions:**

- Test environment with a peer SEPP instance (as cSEPP), which may be simulated.
- The SEPP under test and the peer SEPP have mutually authenticated and already established N32-c connection.
- Exchanging of Data-type encryption policies and Modification policies is required to be performed between the SEPP under test and the peer SEPP.
- The tester shall have access to the interfaces of the SEPP under test and the peer SEPP.
- The tester has configured on the SEPP under test the policies for receiving messages, i.e. the Data-type encryption policy  $d$  of the peer SEPP and the Modification policy  $m$  for the peer SEPP and an IPX provider  $I$  used for the peer SEPP.
- The tester has configured on the peer SEPP the policies for sending, i.e. the peer SEPP's Data-type encryption policy  $d'$  and the Modification policy  $m'$  for the IPX provider  $I$  used for the peer SEPP.
- There are three cases to test:
  - a) the data encryption policies  $d$  and  $d'$  are identical, the modification policies  $m$  and  $m'$  are different
  - b) the data encryption policies  $d$  and  $d'$  are different, the modification policies  $m$  and  $m'$  are identical
  - c) both the data encryption policies  $d$  and  $d'$  and the modification policies  $m$  and  $m'$  are different

**NOTE:** The test case below only applies in case the SEPP under test supports manual configuration of the data encryption policy and/or modification policy for the specific roaming partner and IPX provider.

- The tester has configured on SEPP under test the action to be taken for policy mismatch, which is sending error message.

**Execution Steps:**

For each of the three cases above, the following is executed:

1. The peer SEPP sends a Security Parameter Exchange Request message to the SEPP under test including the peer SEPP's Data-type encryption policy  $d'$ , and the Modification policy  $m'$ .
2. The SEPP under test stores the received Data-type encryption policy  $d'$  and the Modification policy  $m'$ , then compare them with the Data-type encryption policy  $d$  and the Modification policy  $m$  configured on it.

**Expected Results:**

The SEPP under test sends an error signalling message to the peer SEPP on the N32-c connection or logs the error.

**Expected format of evidence:**

Logs and the communication flow saved in a .pcap file.

#### 4.2.2.7 JWS profile restriction

*Requirement Name:* JWS profile restriction

*Requirement Reference:* TS 33.501 [3], clause 13.2.4.9

*Requirement Description:*

SEPPs and IPXs follow the JWS profile as defined in TS 33.210 [8] with the restriction that they shall only use ES256 algorithm.

*Threat References:* TR 33.926 [4], clause G.2.4.1, Use of weak JWS algorithm

*Test case:*

**Test Name:** TC\_JWS\_PROFILE\_RESTRICTION

**Purpose:**

Verify that the SEPP under test is able to restrict the JWS profile to only use ES256 algorithm with IPX entities.

**Procedure and execution steps:**

**Pre-Conditions:**

- Network product documentation of the SEPP under test, containing the information about the supported signature algorithms for JWS operation.
- Test environment with a peer SEPP instance, which may be simulated.
- Test environment with one node simulating an IPX-provider, which supports JWS operation among others.
- The SEPP under test and the peer SEPP have mutually authenticated and already established N32-c connection.
- The tester shall have access to the interfaces of the SEPP under test, the peer SEPP, and the simulated IPX node.
- The tester has configured both the SEPP under test and peer SEPP for N32-f communication via the simulated IPX node.
- The tester has configured a JWS profile differently from what is required in TS 33.501 [3], clause 13.2.4.9 in the simulated IPX node for JWS operation.

**Execution Steps:**

1. The tester shall check that the supported JWS algorithms in the network product documentation complies with the requirement on the restriction.
2. The tester sends a N32-f message from the peer SEPP via the intermediate IPX node towards the SEPP under test.
3. The IPX node modifies one or more attributes of the N32-f message from the peer SEPP and creates a modifiedDataToIntegrityProtect object, which is protected by the IPX node using the JWS algorithm configured by the tester.
4. The IPX node forwards the modified N32-f message to the SEPP under test.
5. Based on the internal log files, the tester validates how the received N32-f message is handled by the SEPP under test.

**Expected Results:**

The modified N32-f message from the IPX node is discarded by the SEPP under test.

**Expected format of evidence:**

Logs and the communication flow saved in a .pcap file.

#### 4.2.2.8      No misplacement of encrypted IEs in JSON object by IPX

*Requirement Name:* No misplacement of encrypted IE in JSON object by IPX

*Requirement Reference:* TS 33.501 [3], clause 13.2.3.4 and clause 13.2.4.1

*Requirement Description:*

As specified in TS 33.501 [3] clause 13.2.3.4, the following basic validation rules are always applied irrespective of the policy exchanged between two roaming partners:

- IEs requiring encryption will not be inserted at a different location in the JSON object.

A SEPP verifies that an intermediate IPX has not moved or copied an encrypted IE to a location that would be reflected from the producer NF in an IE without encryption" as specified in TS 33.501 [3], clause 13.2.4.1.

*Threat References:* TR 33.926 [4], clause G.2.4.2 Exposure of confidential IEs in N32-f message

*Test case:*

**Test Name:** TC\_NO\_ENCRYPTED\_IE\_MISPLACEMENT

**Purpose:**

Verify that the SEPP under test is able to verify that an intermediate IPX has not misplaced (moved or copied) an encrypted IE to a different location in a JSON object that would be reflected from the producer NF for an IE without encryption.

**Procedure and execution steps:**

**Pre-Conditions:**

- System documentation of the SEPP under test, which details how raw public keys/certificates of peer SEPPs are to be configured and how internal log files can be accessed.
- A second SEPP instance for N32 communication with the SEPP under test, which allows for the creation of custom N32-f messages. This system may be simulated.
- Both SEPPs are to be configured with a raw public key/certificate of their communication peer to be able to establish a N32-c connection.
- Test environment with one node simulating an IPX-provider. This functionality includes parsing N32-f messages, creation of JSON-patches for message modifications and JWS operations, among others. It is configured with a modification policy.
- An arbitrary Data-type encryption policy which includes at least one information element requiring encryption on N32-f.
- The SEPP under test is to be configured with the Data-type encryption policy and the same modification policy as the one configured on the simulated IPX-system.

**Execution Steps:**

1. Both SEPPs are configured for N32-f communication via the simulated IPX-system.
2. Both SEPPs establish a mutual N32-c connection.
3. The tester sends a N32-f message from the secondary SEPP via the IPX-system towards the SEPP under test. This message needs to contain at least one information element that requires encryption according to the locally configured Data-type encryption policy.
4. The IPX-system modifies the N32-f message according to its configured modification policy. The tester then inserts the encBlockIDx into a cleartext IE in the modified N32-f message before sending to the SEPP under test.
5. The IPX-system sends the modified N32-f message to the SEPP under test.
6. Based on the internal log files, the tester validates how the received N32-f message is handled by the SEPP under test.

**Expected Results:**

- The N32-f message is discarded by the SEPP under test. The error defined in the clause 6.1.5.3.7 of TS 29.573[6] is sent by the SEPP via N32-c interface.

**Expected format of evidence:**

Logs and the communication flow saved in a .pcap file.

#### 4.2.2.9            Correct Handling of Inter-PLMN Routing

*Requirement Name:* Correct Handling of Inter-PLMN Routing

*Requirement Reference:* TS 29.500 [7], clause 6.1.4.3.3

*Requirement Description:*

If the SEPP receives an HTTP request from a NF with a request URI containing a telescopic FQDN and with a 3gpp-Sbi-Target-apiRoot header, the SEPP ignores the 3gpp-Sbi-Target-apiRoot header and route the request using the telescopic FQDN.

*Threat References:* TR 33.926 [4], clause G.2.x.a, Inter-PLMN routing using the incorrect reference

*Test Case:*

**Test Name:** TC\_CORRECT\_INTER\_PLMN\_ROUTING

**Purpose:**

Verify that the SEPP under test correctly route the NF request to a remote PLMN when receiving both a 3gpp-Sbi-Target-apiRoot header and a telescopic FQDN contained in the Request URI in the HTTP request from a NF.

**Procedure and execution steps:**

**Pre-Conditions:**

- System documentation of the SEPP under test, which details the methods supported for TLS protection between the NF and the SEPP and how internal log files can be accessed.
- A peer SEPP instance of a remote PLMN for N32 communication with the SEPP under test, which may be simulated.
- A NF for sending HTTP request to the remote PLMN of the peer SEPP via the SEPP under test, which may be simulated and supports both telescopic FQDN and the custom 3gpp-Sbi-Target-apiRoot header. The NF is configured with:
  - The NF service profile containing service URI with "https" scheme and an authority of the remote PLMN for communication with the NF producer in the remote PLMN.
  - The telescopic FQDN of the NF producer in the remote PLMN, having the FQDN of the SEPP under test as the trailing part.
  - The FQDN of the SEPP under test.
- The SEPP under test is configured with:
  - The FQDN of the peer SEPP in the remote PLMN.
  - The security mechanism negotiated with the peer SEPP in the remote PLMN.

**Execution Steps**

- 1) The NF sets up a TLS connection with the authoritative server for the configured telescopic FQDN, i.e. the SEPP under test.
- 2) The NF sends a HTTP service request with the request URI containing the configured telescopic FQDN within the TLS connection to the SEPP under test, before which the tester inserts in the HTTP request a 3gpp-Sbi-Target-apiRoot header set to the apiRoot of a NF producer in another PLMN different from the remote PLMN.
- 3) The NF sends a HTTP service request within the TLS connection to the SEPP under test, before which the tester inserts in the HTTP request a 3gpp-Sbi-Target-apiRoot header set to the apiRoot of the NF producer in the remote PLMN and changes the telescopic FQDN in request URI to be different from the configured one.

**Expected Results:**

After step 2), the peer SEPP received the HTTP request from the NF through the SEPP under test.

After step 3), the peer SEPP did not receive the HTTP request from the NF through the SEPP under test

#### **Expected format of evidence:**

Evidence suitable for the interface, e.g. screenshot containing the operational results.

#### **4.2.2.10            Correct Handling of Custom HTTP Header with PRINS Security**

*Requirement Name:* Correct Handling of the Custom HTTP Header with PRINS Security

*Requirement Reference:* TS 29.500 [7], clause 6.1.4.3.4

#### *Requirement Description:*

The 3gpp-Sbi-Target-apiRoot header is not used between SEPPs if PRINS security is negotiated between the SEPPs.

*Threat References:* TR 33.926 [4], clause G.2.x.b, Tampering of target API root

#### *Test Case:*

**Test Name:** TC\_HANDLING\_CUSTOM\_HTTPHEADER\_WITH\_PRINS

#### **Purpose:**

Verify that the SEPP under test correctly handle the 3gpp-Sbi-Target-apiRoot custom HTTP header received from a NF when PRINS security is negotiated with the peer SEPP in a remote PLMN.

#### **Procedure and execution steps:**

##### **Pre-Conditions:**

- System documentation of the SEPP under test, including the security mechanisms supported for protection between SEPPs.
- A peer SEPP instance of a remote PLMN for N32 communication with the SEPP under test, which may be simulated.
- A NF for sending HTTP request to the remote PLMN of the peer SEPP via the SEPP under test, which may be simulated and supports 3gpp-Sbi-Target-apiRoot header. The NF is configured to route all HTTP messages with inter PLMN FQDN as the "authority" part of the URI via the SEPP under test.
- The SEPP under test is configured with PRINS security as the security mechanism negotiated with the peer SEPP in the remote PLMN.
- A TLS connection is setup between the SEPP under test and the peer SEPP in the remote PLMN for N32-f forwarding.

##### **Execution Steps**

- 1) The NF initiates a HTTP message sent to the SEPP under test, which includes the 3gpp-Sbi-Target-apiRoot header containing the apiRoot of the target URI in the remote PLMN and the apiRoot in the request URI set to the apiRoot of the SEPP under test.
- 2) The SEPP under test forwards the HTTP request to the peer SEPP in the remote PLMN within the N32-f TLS tunnel.

#### **Expected Results:**

The peer SEPP received the protected HTTP Request from the NF through the SEPP under test, in which the apiRoot in the request URI is the apiRoot of the target URI in the remote PLMN and no 3gpp-Sbi-Target-apiRoot header is present.

#### **Expected format of evidence:**

Evidence suitable for the interface, e.g. screenshot containing the operational results.

## 4.2.3 Technical Baseline

### 4.2.3.1 Introduction

The present clause provides baseline technical requirements.

### 4.2.3.2 Protecting data and information

#### 4.2.3.2.1 Protecting data and information – general

There are no SEPP-specific additions to clause 4.2.3.2.1 of TS 33.117 [2].

#### 4.2.3.2.2 Protecting data and information – unauthorized viewing

There are no SEPP-specific additions to clause 4.2.3.2.2 of TS 33.117 [2].

#### 4.2.3.2.3 Protecting data and information in storage

There are no SEPP-specific additions to clause 4.2.3.2.3 of TS 33.117 [2].

#### 4.2.3.2.4 Protecting data and information in transfer

There are no SEPP-specific additions to clause 4.2.3.2.4 of TS 33.117 [2].

#### 4.2.3.2.5 Logging access to personal data

There are no SEPP-specific additions to clause 4.2.3.2.5 of TS 33.117 [2].

### 4.2.3.3 Protecting availability and integrity

There are no SEPP-specific additions to clause 4.2.3.3 of TS 33.117 [2].

### 4.2.3.4 Authentication and authorization

There are no SEPP-specific additions to clause 4.2.3.4 of TS 33.117 [2].

### 4.2.3.5 Protecting sessions

There are no SEPP-specific additions to clause 4.2.3.5 of TS 33.117 [2].

### 4.2.3.6 Logging

There are no SEPP-specific additions to clause 4.2.3.6 of TS 33.117 [2].

## 4.2.4 Operating Systems

There are no SEPP-specific additions to clause 4.2.4 of TS 33.117 [2].

## 4.2.5 Web Servers

*Requirement Name:* Correct Handling of HTTPS Trust Anchoring

*Requirement Reference:* TS 33.501 [3], clause 13.1.2

*Requirement Description:* The SEPP maintains a set of trust anchors, each consisting of a list of trusted root certificates and a list of corresponding PLMN-IDs. Any given PLMN-ID appears in at most one trust anchor. During N32-c connection setup, the SEPP maps the PLMN-ID of the remote SEPP leaf (server or client) certificate to the associated trust anchor for the purposes of certificate chain verification. Only the root certificates in the associated list is treated as trusted during certificate chain verification. If the remote SEPP certificate contains multiple PLMN-IDs that are mapped to different trust anchors, then that certificate is rejected.

*Threat References:* TR 33.926 [4], clause 5.3.3.7, Eavesdropping, clause 5.3.6.10, Insecure Network Services

*Test Case:*

**Test Name:** TC\_CORRECT\_TRUST\_ANCHORING**Purpose:**

Verify that the SEPP under test correctly implements trust anchoring when setting up HTTPS (TLS) connections.

**Procedure and execution steps:****Pre-Conditions:**

- System documentation of the SEPP under test, which details the methods supported for TLS protection on N32 and how internal log files can be accessed.
- Two simulated, independent root CAs, denoted RCA1 and RCA2
- A peer SEPP instance of a remote PLMN for N32 communication with the SEPP under test, which may be simulated.

**Execution Steps**

1. The tester selects two different PLMN IDs, denoted ID1 and ID2, e.g. ID1={MCC=262, MNC=01} and ID2={MCC=262, MNC=02}.
2. The tester configures two trust anchors in the SEPP under test, as follows:
  - ID1 gets associated with RCA1 and
  - ID2 gets associated with RCA2.
3. The tester generates a server certificate for the SEPP under test with which it authenticates itself towards a peer SEPP. The peer SEPP is configured to accept these certificates.
4. The tester generates four TLS client certificates C1, C2, C3, C4 for the peer SEPP, as follows:
  - C1 contains ID1 and is signed by RCA1 (or a subCA which is signed only by RCA1)
  - C2 contains ID2 and is signed by RCA2 (or a subCA which is signed only by RCA2)
  - C3 contains ID1 and is signed by RCA2 (or a subCA which is signed only by RCA2)and
  - C4 contains both ID1 and ID2 and is signed by RCA1 or RCA2 (or a subCA which is signed by RCA1 or RCA2).

NOTE 1: The expression "contains IDX" means that there exists a Subject Alternative Name (SAN) field in the certificate that contains the value chosen in step 1. Examples of the SAN field contents are example.sepp.5gc.mnc02mcc262.3gppnetwork.org and example.sepp.5gc.mnc01mcc262.exampleipx.ipxnetwork.org

5. The peer SEPP is configured to authenticate itself using C1. If C1 was issued by a SubCA, then the SubCA certificate is included in the certificate chain which the peer SEPP uses to authenticate itself. The tester initiates an N32c connection from the SEPP under test towards the peer SEPP and observes whether the HTTPS connection succeeds, and, if not, documents the failure reasons as shown in the log files of the SEPP under test.
6. The tester repeats step 5, replacing C1 with C2, C3, and C4 iteratively.
7. Steps 3 to 6 are repeated in order to test for reversed client/server roles. Specifically, in step 3 the tester generates a client certificate (instead of a server certificate), in step 4 the tester generates four server certificates (instead of four client certificates), and in step 5 the tester initiates the connection from the peer SEPP (instead of initiating the connection from the SEPP under test).

**Expected Results:**

In step 5, the TLS (HTTPS) connection setup succeeds for the iterations with C1 and C2 and fails for the iterations with C3 and C4.

NOTE 2: The iteration with C3 fails because the PLMN-ID indicated in the client certificate does not match any of the trusted certificates in the corresponding trust anchor. The iteration with C4 fails because the PLMN-IDs indicated in the client certificate are not associated with the same trust anchor.

#### **Expected format of evidence:**

The evidence is in the form of log file entries associated with eight TLS connection establishment attempts, where entries for the cases with C1 and C2 indicate success, and entries for the cases with C3 and C4 indicate failure and show the corresponding failure reasons, including TLS Error Alert 48 (unknown\_ca).

### **4.2.6 Network Devices**

There are no SEPP-specific additions to clause 4.2.6 in TS 33.117 [2].

## **4.3 SEPP-specific adaptations of hardening requirements and related test cases**

### **4.3.1 Introduction**

The requirements proposed hereafter (with the relative test cases) aim to securing SEPP by reducing its surface of vulnerability. In particular, the identified requirements aim to ensure that all the default configurations of SEPP (including operating system software, firmware and applications) are appropriately set.

### **4.3.2 Technical baseline**

There are no SEPP-specific additions to clause 4.3.2 in TS 33.117 [2].

### **4.3.3 Operating systems**

There are no SEPP-specific additions to clause 4.3.3 in TS 33.117 [2].

### **4.3.4 Web servers**

There are no SEPP-specific additions to clause 4.3.4 in TS 33.117 [2].

### **4.3.5 Network devices**

There are no SEPP-specific additions to clause 4.3.5 in TS 33.117 [2].

### **4.3.6 Network functions in service-based architecture**

There are no SEPP-specific additions to clause 4.3.6 in TS 33.117 [2].

## **4.4 SEPP-specific adaptations of basic vulnerability testing requirements and related test cases**

### **4.4.1 Introduction**

There are no SEPP specific additions to clause 4.4.1 of TS 33.117 [2].

### **4.4.2 Port Scanning**

There are no SEPP specific additions to clause 4.4.2 of TS 33.117 [2].

### **4.4.3 Vulnerability scanning**

There are no SEPP specific additions to clause 4.4.3 of TS 33.117 [2].

#### 4.4.4 Robustness and fuzz testing

The test cases under clause 4.4.4 of TS 33.117 [2] are applicable to SEPP.

The interface defined for the SEPP are in 4.2.4 of TS 23.501 [8].

According to clause 4.4.4 of TS 33.117 [2], the transport protocols available on the interfaces providing IP-based protocols need to be robustness tested. Following TCP/IP layer model and considering all the protocols over transport layer, for SEPP, the following interfaces and protocols are in the scope of the testing:

- For SBI: the TCP, HTTP2 and JSON protocols.
- For N32-C: The TCP, HTTP2 and JSON protocols.
- For N32-F: The TCP, HTTP2 and JSON protocols.

NOTE: There could be other interfaces and/or protocols requiring testing under clause 4.4.4 of TS 33.117 [2]

---

## Annex A (informative): Change history

Change history							
date	Meeting	Tdoc	CR	Rev	Cat	Subject/Comment	New version
2019-09	SA#85					Change control version	16.0.0
2019-10						EditHelp review	16.0.1
2019-12	SA#86	SP-191138	0001	1	F	Adding abbreviations and corrections for alignment	16.1.0
2021-03	SA#91e	SP-210117	0005	1	F	Protection policies – TBD updated	16.2.0
2021-03	SA#91e	SP-210117	0006	1	F	Protection policies test case	16.2.0
2021-03	SA#91e	SP-210117	0007	1	F	Clarification on confidential IEs replacement handling in original N32-f message	16.2.0
2021-06	SA#92e	SP-210446	0008	-	F	Resolving EN on the error handling	16.3.0
2021-06	SA#92e	SP-210440	0009	-	B	CR to include R-16 feature of SEPP to 33.517	17.0.0
2023-06	SA#100	SP-230677	0010	2	B	New SCAS test on trust anchoring	18.0.0
2023-06	SA#100	SP-230677	0011	1	B	Robustness interfaces and protocols defined for SEPP	18.0.0
2023-06	SA#100	SP-230677	0012	1	F	SCAS release reference corrections	18.0.0
2025-07	SA#108	SP-250657	0013	-	F	Addition of test name and clean up of 33.517	19.0.0