ETSI TS 133 517 V16.2.0 (2021-04)

5G;
5G Security Assurance Specification (SCAS)
for the Security Edge Protection Proxy (SEPP)
network product class
(3GPP TS 33.517 version 16.2.0 Release 16)

Reference
RTS/TSGS-0333517vg20

Keywords
5G,SECURITY

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd 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 (Release 15): "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".

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, an SEPP shall satisfy the following:

- It is assumed for the purpose of the present SCAS that an 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 33.50 [3]1 or are referred to from TS 33.501 [3] Security procedures pertaining to an 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 shall be 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 shall verify 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 shall check 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:

" 1.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 shall point 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:

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 shall compare 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 shall perform one of the following actions, according to operator policy:

- Send the error message as specified in TS 29.573 [73], 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 shall follow the JWS profile as defined in TS 33.210 [3] 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:

"The following basic validation rules shall always be applied irrespective of the policy exchanged between two roaming partners:

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

as specified in TS 33.501 [3], clause 13.2.3.4.

"A SEPP shall verify 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 encrypted information element 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.

Editor's Note: the result needs to be aligned with the relevant error handling description to be added in TS 33.501.

Expected format of evidence:

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

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

There are no SEPP-specific additions to clause 4.2.5 of TS 33.117 [2].

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

There are no SEPP-specific additions to clause 4.4 in 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	SP-190702				Presented for information and approval	1.0.0
2019-09	SA#85	SP-190910				Revised to include the right version of the draft	1.0.1
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

History

Document history		
V16.1.0	October 2020	Publication
V16.2.0	April 2021	Publication