**3GPP TSG-SA3 Meeting #111 *S3-23xxxx***

**Berlin, Germany, 22 - 26 May 2023**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **33.517** | **CR** | **XXXX** | **rev** | **-** | **Current version:** | **17.0.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | SCAS release reference corrections | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Huawei, HiSilicon | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SCAS\_5G\_Ph2 | | | | |  | ***Date:*** | | | 2023-05-22 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | | Rel-18 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | SA3 has been adding the release numbers explicitly to any of the references pertaining to the network function targeted by the SCAS work, for example reference 2 in TS 33.511. This is because the SCAS work has always been one "release late" since it is challenging to develop the SCAS requirements and tests in parallel to targeted new features within the same release timeline. The references have not been regularly updated and some SCAS specifications include more than one reference to the same specification, for example references 2 and 7 in TS 33.512. This practice is neither future proof nor it is documented anywhere. Furthermore, for SCAS evaluation of network products, this dependency on previous releases in SCAS documents turned out to be not very useful anyway. This issue has been discussed several times in previous SA3 meetings and the proposed resolution is documented in [S3-231050](https://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_110_Athens/docs/S3-231050.zip). | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Removal of the release number from the relevant references and minor reformulations to avoid verbatim content copies from other specifications | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Unnecessary dependencies on previous releases and risk for confusion on scope of SCAS specifications | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 4.2.2.2, 4.2.2.4, 4.2.2.5, 4.2.2.6, 4.2.2.7, 4.2.2.8, 4.2.2.9, 4.2.2.10 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

\*\*\*\* Start of Changes\*\*\*\*

# 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".

[x] 3GPP TS 33.210: "Network Domain Security (NDS); IP network layer security"

\*\*\*\* Next Changes\*\*\*\*

#### 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 expected to 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.

\*\*\*\* Next Changes\*\*\*\*

#### 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 is expected to 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 is expected to 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.

\*\*\*\* Next Changes\*\*\*\*

#### 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 is expected to 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.

\*\*\*\* Next Changes\*\*\*\*

#### 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 is expected to 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 is expected to perform 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.

\*\*\*\* Next Changes\*\*\*\*

#### 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 are expected to follow the JWS profile as defined in TS 33.210 [x] 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.

\*\*\*\* Next Changes\*\*\*\*

#### 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 expected to 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.

A SEPP is expected to 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 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.

\*\*\*\* Next Changes\*\*\*\*

##### 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 is expected to ignore 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 receving 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.

\*\*\*\* Next Changes\*\*\*\*

##### 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 expected to be 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.

\*\*\*\* End of Changes\*\*\*\*