**3GPP TSG-CT4 Meeting #101-e C4-205495**

**E-Meeting, 3rd Nov 2020 - 13th Nov 2020** *Revision of 5016*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **29.573** | **CR** | **0049** | **rev** | **1** | **Current version:** | **16.4.0** |  |
|  | | | | | | | | |
| *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 | **x** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | PLMN ID handling over N32 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** | CT4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** |  | | | | |  | ***Date:*** | | |  |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | |  |
|  | *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:*** | | Clause 5.2.4 N32-f Context Termination Procedure specifies that the Security Capability Negotiation and/or the Parameter Exchange procedures establish N32-f Contexts between the two SEPP peers. 3GPP TS 33.501 clause 5.9.3 specifies the following Rel-15 requirement: "The receiving SEPP shall be able to verify whether the sending SEPP is authorized to use the PLMN ID in the received N32 message. This requirements needs to be referenced in TS 29.573.  Another matter is, in order to address a use case when a visited network uses multiple PLMN IDs, c-SEPP will need to inform the p-SEPP about this. This is necessary to enable p-SEPP to correctly do the checking (see the above requirement) in such a scenario. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | This CR addresses the following two issues.  Clarifies the existing Rel-15 security requirement:   * Clause 5.2.2. Missing "sender FQDN" and "SecNegotiateRspData" IEs are added. * Clause 5.2.3.2. Missing "sender FQDN" is added. * Clause 5.3.2.1. Reference to clause 5.9.3 in 3GPP TS 33.501 is added. * Clause 5.3.2.4. Reference to clause 5.3.2.1 is added, which specifies unsuccessful processing of a message.   Adds optional new feature, which enables c-SEPP to send multiple FQDNs to a p-SEPP:   * Clauses 5.2.2 and 5.2.3.2. FQDN(s) are added to requests. * Clause 6.1.5.2.2. New optional attribute 'additionalFqdns' is added to SecNegotiateReqData data type. * Clause 6.1.5.2.4. New optional attribute 'additionalFqdns' is added to SecParamExchReqData data type. * Annex A.2 updated in line with the above. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Ambiguity in the spec. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.2.2, 5.2.3.2, 5.3.2.1, 5.3.2.4, 6.1.5.2.2, 6.1.5.2.4, A.2. | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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 adds backward compatible corrections to Open API. | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | | Rev1:   * Cover sheet fixed. * 5.2.2. * 5.2.3.2. * 5.3.2.1. * 6.1.5.2.2. * 6.1.5.2.4. * A.2. | | | | | | | | |

\* \* \* First Change \* \* \* \*

### 5.2.2 Security Capability Negotiation Procedure

The initiating SEPP shall initiate a Security Capability Negotiation procedure towards the responding SEPP to agree on a security mechanism to use for protecting NF service related signalling over N32-f. An end to end TLS connection shall be setup between the SEPPs before the initiation of this procedure. The procedure is described in Figure 5.2.2-1 below.



Figure 5.2.2-1: Security Capability Negotiation Procedure

1. The initiating SEPP issues a HTTP POST request towards the responding SEPP with the request body containing the "SecNegotiateReqData" IE carrying the following information:

- Supported security capabilities (i.e PRINS and/or TLS);

- Whether the 3gpp-Sbi-Target-apiRoot HTTP header is supported, if TLS security is supported;

- Sender FQDN(s).

2a. On successful processing of the request, the responding SEPP shall respond to the initiating SEPP with a "200 OK" status code and a POST response body that contains "SecNegotiateRspData" IE carrying the following information:

- Selected security capability (i.e PRINS or TLS);

- Whether the 3gpp-Sbi-Target-apiRoot HTTP header is supported, if TLS security is selected;

- Sender FQDN(s).

The responding SEPP compares the initiating SEPP's supported security capabilities to its own supported security capabilities and selects, based on its local policy, a security mechanism, which is supported by both the SEPPs. If the selected security capability indicates any other capability other than PRINS, then the HTTP/2 connection initiated between the two SEPPs for the N32 handshake procedures shall be terminated. The negotiated security capability shall be applicable on both the directions. If the selected security capability is PRINS, then the two SEPPs may decide to create (if not available) / maintain HTTP/2 connection(s) where each SEPP acts as a client towards the other (which acts as a server). This may be used for later signalling of N32-f error reporting procedure (see clause 5.2.5) and N32-f context termination procedure (see clause 5.2.4).

2b. On failure, the responding SEPP shall respond to the initiating SEPP with an appropriate 4xx/5xx status code as specified in clause 6.1.4.2.

\* \* \* 2nd Change \* \* \* \*

#### 5.2.3.2 Parameter Exchange Procedure for Cipher Suite Negotiation

The parameter exchange procedure for cipher suite negotiation shall be performed after the security capability negotiation procedure if the selected security policy is PRINS.

The procedure is described in Figure 5.2.3.2-1 below.



Figure 5.2.3.2-1: Parameter Exchange Procedure for Cipher Suite Negotiation

1. The initiating SEPP issues a HTTP POST request towards the responding SEPP with the request body containing the "SecParamExchReqData" IE carrying the following information

- Supported cipher suites;

- Sender FQDN(s).

The supported cipher suites shall be an ordered list with the cipher suites mandated by 3GPP TS 33.501 [6] appearing at the top of the list.

The initiating SEPP also provides a N32-f context identifier for the responding SEPP to use towards the initiating SEPP for subsequent JOSE Protected Message Forwarding procedures over N32-f (see clause 5.3.3) when the responding SEPP acts as the forwarding SEPP.

2a. On successful processing of the request, the responding SEPP shall respond to the initiating SEPP with a "200 OK" status code and a POST response body that contains the following information

- Selected cipher suite;

- Sender FQDN.

The responding SEPP compares the initiating SEPP's supported cipher suites to its own supported cipher suites and selects, based on its local policy, a cipher suite, which is supported by both the SEPPs. The responding SEPP's supported cipher suites shall be an ordered list with the cipher suites mandated by 3GPP TS 33.501 [6] appearing at the top of the list. The selected cipher suite is applicable for both the directions of communication between the SEPPs.

The responding SEPP also provides a N32-f context identifier for the initiating SEPP to use towards the responding SEPP for subsequent JOSE Protected Message Forwarding procedures over N32-f (see clause 5.3.3) when the initiating SEPP acts as the forwarding SEPP.

2b. On failure, the responding P-SEPP shall respond to the initiating SEPP with an appropriate 4xx/5xx status code as specified in clause 6.1.4.3.

\* \* \* 3rd Change \* \* \* \*

### 5.3.2 Use of Application Layer Security

#### 5.3.2.1 General

If the negotiated security capability between the two SEPPs is PRINS, one or more HTTP/2 connections between the two SEPPs for the forwarding of JOSE protected message shall be established, which may involve IPX providers on path. The forwarding of messages over the N32-f interface involves the following steps at the sending SEPP:

1. Identification of the protection policy applicable for the API being invoked (i.e either a request/response NF service API or a subscribe/unsubscribe service API or a notification API).

2. Message reformatting as per the identified protection policy.

3. Forwarding of the reformatted message over the N32 interface.

The processing of a message received over the N32-f interface at the receiving SEPP involves the following steps.

1. Identify the N32-f context using the N32-f context Id received in the message.

2. Verify the integrity protection of the message using the keying material obtained from the TLS layer during the parameter exchange procedure for that N32-f context (see 3GPP TS 33.501 [6]). The TLS connection from which the keying material is obtained is the N32-c TLS connection used for the parameter exchange procedure.3. Decrypt the ciphertext part of the received JWE message. Decode the "aad" part of the JWE message using BASE64URL decoding.

4. Form the original JSON request / response body from the decrypted ciphertext and the decoded integrity verified "aad" block.

5. For each entry in the "modificationsBlock" of the received message:

- First verify the integity protection of that entry using the keying material applicable for the IPX that inserted that block (using the "identity" IE in the "modificationsBlock");

- Identify the modifications policy exchanged during the parameter exchange procedure with the sending SEPP if the IPX that inserted the modificationsBlock is from the sending SEPP side; else identify the modifications policy applicable for the IPX based on local configuration;

- Check if the inserted modifications are as per the identified modifications policy;

- Apply the modifications as a JSON patch over the formed original JSON request / response body from step 4.

6. If the reconstructed HTTP message has a "Authorization" header, then the SEPP shall check whether the service consumer's PLMN ID is present in the Bearer token contained in the Authorization header (see 3GPP TS 29.510 [18], clause 6.3.5.2.4) and if it matches with the "Remote PLMN ID" of the N32-f context. If they do not match, the SEPP shall respond to the sending SEPP with "403 Forbidden" status code with the application specific cause set as "PLMNID\_MISMATCH".

NOTE 1: In this case, the N32-f Error Reporting procedure specified in clause 5.2.5 is not used since the processing of the complete N32-f message fails at the receiving SEPP.

NOTE x: If the service consumer's PLMN ID is present in the reconstructed HTTP message, then the receiving SEPP compares this with the sending SEPP's PLMN ID, which is retrieved from N32f Context (see clause 5.9.3 in 3GPP TS 33.501 [6]). See the above step 6 for the receiving SEPP behaviour. If the service consumer's PLMN ID is not present, the comparison is not done.

\* \* \* 4th Change \* \* \* \*

#### 5.3.2.4 Message Forwarding to Peer SEPP

Once a SEPP reformats the HTTP/2 message into the "N32ReformattedReqMsg"/"N32ReformattedRspMsg" JSON object as specified in clause 5.3.2, the SEPP forwards the message to the receiving SEPP by invoking a HTTP POST method as shown in figure 5.3.2.4-1 below.



Figure 5.3.2.4-1 Message Forwarding between SEPP on N32-f

1. The initiating SEPP issues a HTTP POST request towards the responding SEPP with the request body containing the "N32ReformattedReqMsg" IE carrying the reformatted HTTP/2 message. The request message shall contain the "n32fContextId" information provided by the responding SEPP to the initiating SEPP earlier during the parameter exchange procedure (see clause 5.2.3). The responding SEPP shall use the "n32fContextId" information to:

- Locate the agreed cipher suite and protection policy;

- Locate the n32ContextId to be used in the response.

2a. On successful processing of the request, the responding SEPP shall:

- reconstruct the HTTP/2 message towards the NF service producer;

- forward the reconstructed HTTP/2 message to the NF service producer;

- wait for the response from the NF service producer; and then

- once the response from the NF service producer is received, respond to the initiating SEPP with a "200 OK" status code and a POST response body that contains the "N32ReformattedRspMsg". The "N32ReformattedRspMsg" shall contain the reformatted HTTP response message from the responding PLMN. The response message shall contain the "n32fContextId" information provided by the initiating SEPP to the responding SEPP earlier during the parameter exchange procedure (see clause 5.2.3).

NOTE x: For unsuccessful processing of the request see clause 5.3.2.1.

The responding SEPP shall be able to map the response received from the NF service producer to the HTTP/2 stream ID for the corresponding response it needs to generate towards the initiating SEPP. The HTTP/2 stream ID and the HTTP/2 connection information on either side shall be used to derive this mapping.

2b. On failure, the responding SEPP shall respond to the initiating SEPP with an appropriate 4xx/5xx status code as specified in clause 6.2.4.2.

\* \* \* 5th Change \* \* \* \*

##### 6.1.5.2.2 Type: SecNegotiateReqData

Table 6.1.5.2.2-1: Definition of type SecNegotiateReqData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| sender | Fqdn | M | 1 | This IE shall uniquely identify the SEPP that is sending the request. This IE is used to store the negotiated security capability against the right SEPP. |
| supportedSecCapabilityList | array(SecurityCapability) | M | 1..N | This IE shall contain the list of security capabilities that the requesting SEPP supports. |
| 3GppSbiTargetApiRootSupported | boolean | C | 0..1 | This IE should be present and indicate that the 3gpp-Sbi-Target-apiRoot HTTP header is supported, if TLS security is supported for N32f message forwarding.  When present, it shall indicate if TLS security using the 3gpp-Sbi-Target-apiRoot HTTP header is supported:  - true: supported  - false (default): not supported |
| additionalFqdnList | array(Fqdn) | O | 1..N | A list of FQDNs associated with the SEPP, which is sending the request. A map (list of key-value pairs where FQDN string serves as key to the PLMN ID value) to be stored by the receiving SEPP in a N32-f Context (see clause 5.9.3 in 3GPP TS 33.501 [6],). See NOTE 1. |
| NOTE 1: FQDN may or may not contain PLMN ID. If FQND does not contain PLMN ID, then the receiving SEPP needs to be configured to resolve such FQDN to one or more PLMN ID(s). | | | | |

\* \* \* 6th Change \* \* \* \*

##### 6.1.5.2.4 Type: SecParamExchReqData

Table 6.1.5.2.4-1: Definition of type SecParamExchReqData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| n32fContextId | string | M | 1 | This IE shall contain the context identifier to be used by the responding SEPP for subsequent JOSE protected message forwarding procedure over N32-f towards the initiating SEPP. The initiating SEPP shall use this context identifier to locate the cipher suite and protection policy exchanged and agreed to be used with the responding SEPP, for the message forwarding procedure over N32-f.  The n32fContextId shall encode a 64-bit integer in hexadecimal representation. Each character in the string shall take a value of "0" to "9" or "A" to "F" and shall represent 4 bits. The most significant character representing the 4 most significant bits of the N32-f context Id shall appear first in the string, and the character representing the 4 least significant bit of the N32-f context Id shall appear last in the string.  Pattern: '^[A-Fa-f0-9]{16}$'  Example: "0600AD1855BD6007". |
| jweCipherSuiteList | array(string) | C | 1..N | This IE shall be present during the parameter exchange procedure for cipher suite negotiation (see clause 5.2.3.2). When present, this IE shall contain the ordered list of JWE cipher suites supported by the requesting SEPP. Valid values for the string are as specified in clause 5.1 of IETF RFC 7518 [13]. |
| jwsCipherSuiteList | array(string) | C | 1..N | This IE shall be present during the parameter exchange procedure for cipher suite negotiation (see clause 5.2.3.2). When present, this IE shall contain the ordered list of JWS cipher suites supported by the requesting SEPP. Valid values for the string are as specified in clause 3.1 of IETF RFC 7518 [13]. |
| protectionPolicyInfo | ProtectionPolicy | C | 0..1 | This IE shall be present during the parameter exchange procedure for protection policy exchange (see clause 5.2.3.3). When present, this IE shall contain the protection policy requested by the requesting SEPP. |
| ipxProviderSecInfoList | array(IpxProviderSecInfo) | C | 1..N | This IE includes the list of IPX security information. |
| sender | Fqdn | C | 0..1 | This IE shall be present if the Parameter Exchange request is sent on a different N32-c HTTP connection than the one used to perform the Security Capability Negotiation procedure. It may be present otherwise.  When present, it shall uniquely identify the SEPP that is sending the request. This IE is used to store the exchanged parameters against the right SEPP. |
| additionalFqdnList | array(Fqdn) | O | 1..N | A list of FQDNs associated with the SEPP, which is sending the request. A map (list of key-value pairs where FQDN string serves as key to the PLMN ID value) to be stored by the receiving SEPP in a N32-f Context (see clause 5.9.3 in 3GPP TS 33.501 [6],). See NOTE 1. |
| NOTE 1: FQDN may or may not contain PLMN ID. If FQND does not contain PLMN ID, then the receiving SEPP needs to be configured to resolve such FQDN to one or more PLMN ID(s). | | | | |

\* \* \* 7th Change \* \* \* \*

## A.2 N32 Handshake API

openapi: 3.0.0

info:

version: '1.1.1'

title: 'N32 Handshake API'

description: |

N32-c Handshake Service.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

\*\*\* Skipped for clarity \*\*\*\*

SecNegotiateReqData:

type: object

required:

- sender

- supportedSecCapabilityList

properties:

sender:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

supportedSecCapabilityList:

type: array

items:

$ref: '#/components/schemas/SecurityCapability'

minItems: 1

3GppSbiTargetApiRootSupported:

type: boolean

default: false

additionalFqdnList:

type: object

additionalProperties:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

minProperties: 1

\*\*\* Skipped for clarity \*\*\*\*

SecParamExchReqData:

type: object

required:

- n32fContextId

properties:

n32fContextId:

type: string

pattern: '^[A-Fa-f0-9]{16}$'

jweCipherSuiteList:

type: array

items:

type: string

minItems: 1

jwsCipherSuiteList:

type: array

items:

type: string

minItems: 1

protectionPolicyInfo:

$ref: '#/components/schemas/ProtectionPolicy'

ipxProviderSecInfoList:

type: array

items:

$ref: '#/components/schemas/IpxProviderSecInfo'

minItems: 1

sender:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

additionalFqdnList:

type: object

additionalProperties:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Fqdn'

minProperties: 1

\*\*\* Skipped for clarity \*\*\*\*

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