**3GPP TSG-SA3 Meeting #100e *S3-202046-r2***

**e-meeting, 17 - 28 August 2020**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.0* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **33.501** | **CR** |  | **rev** | **1** | **Current version:** | **16.3.0** |  |
|  | | | | | | | | |
| *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:*** | N32 interface | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Nokia, Nokia Shanghai Bell | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | 5G\_eSBA | | | | |  | ***Date:*** | | |  |
|  |  | | | |  | |  | | |  |
| ***Category:*** |  |  | | | | | ***Release:*** | | | Rel-16 |
|  | *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-12 (Release 12)* *Rel-13 (Release 13) Rel-14 (Release 14) Rel-15 (Release 15) Rel-16 (Release 16)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Alignment with CT4 in line with LS S3-201464 on Misalignment on HTTP connections for N32-c and on N32-f contexts termination.  Replacement of long-lived with short-lived connection is needed.  In 13.2.1 Clarifications on the deployment of PRINS is needed, if no IPX is in between. CT4 is not ruling out to use PRINS also between SEPPs only as deployment, but SA3 should clarify that this is not preferrred, since TLS can do the job in this case  In 13.2.2.1 clarification on N32-c is usage is missing    In 13.2.2.2 step 6, TLS session may be kept open, it should not be mandatory. Also the text suggests that N32-c and N32-f share the same HTTPS connection, therefore changed to say “TLS sessions”  In 13.2.2.4.1 Type reveiving -> receiving and other small typos | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Clarification on deployment added by Note  Clafication on N32-c for security mechanism negotiation and when PRINS is used | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Unclear deployment scenario  N32-c usage unclear | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 3.1, 13.2.1, 13.2.2.1, 13.2.2.2, 13.2.2.3, 13.2.2.4.1, 1.2.2.4.2, 13.2.4.4.1, | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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:*** | | update of S3-201796 | | | | | | | | |

\*\*\*\*\*\*\*\*\*\*\*\* START OF CHANGES

\*\*\*\*\*\*\*\*\*\*\*\* CHANGE 1 (clause 13.2 Application layer security on the N32 interface ) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13.2.1 General

The internetwork interconnect allows secure communication between service-consuming and a service-producing NFs in different PLMNs. Security is enabled by the Security Edge Protection Proxies (SEPPs) of both networks, henceforth called cSEPP and pSEPP respectively. The SEPPs enforce protection policies regarding application layer security thereby ensuring integrity and confidentiality protection for those elements to be protected.

For the usage of application layer security on the N32 interface with PRINS, it is assumed that there are interconnect providers between cSEPP and pSEPP. The interconnect provider, the cSEPP's operator has a business relationship with, is called cIPX, while the interconnect provider, the pSEPP's operator has a business relationship with, is called pIPX. There could be further interconnect providers in between cIPX and pIPX, but they are assumed to be transparent and simply forward the communication.

The SEPPs use JSON Web Encryption (JWE, specified in RFC 7516 [59]) for protecting messages on the N32-f interface, and the IPX providers use JSON Web Signatures (JWS, specified in RFC 7515 [45]) for signing their modifications needed for their mediation services.

For illustration, consider the case where a service-consuming NF sends a message to a service-producing NF. If this communication is across PLMN operators over the N32 interface, as shown in Figure 13.2.1-1 below, the cSEPP receives the message and applies symmetric key based application layer protection, as defined in clause 13.2 of the present document. The resulting JWE object is forwarded to intermediaries. The pIPX and cIPX can offer services that require modifications of the messages transported over the interconnect (N32) interface. These modifications are appended to the message as digitally signed JWS objects which contain the desired changes. The pSEPP, which receives the message from pIPX, validates the JWE object, extracts the original message sent by the NF, validates the signature in the JWS object and applies patches corresponding to the modifications by intermediaries. The pSEPP then forwards the message to the destination NF.

The N32 interface, if used for PRINS, consists of:

- N32-c connection, for managing the N32 interface, and

- N32-f connection, for sending of JWE and JWS protected messages between the SEPPs.

.

****

**Figure 13.2.1-1: Overview of PRINS**

\*\*\*\*\*\*\*\*\*\*\*\* CHANGE 2 (clause 13.2.2 N32-c connection between SEPPs) \*\*\*\*\*\*\*\*\*

13.2.2.1 General

Via N32-c the security mechanism (as described in the procedure in clause 13.5) are negotiated.

When the negotiated security mechanism to use over N32is PRINS (see overview in clause 13.2.1), the SEPPs use the established TLS connection (henceforth referred to as N32-c connection) to negotiate the N32-f specific associated security configuration parameters required to enforce application layer security on HTTP messages exchanged between the SEPPs. A second N32-c connection is established by the receiving SEPP to enable it to not only receive but also send HTTP Requests.

The N32-c connection is further used for the following purposes when PRINS is used:

- Key agreement: The SEPPs independently export keying material associated with the first N32-c connection between them and use it as the pre-shared key for generating the shared session key required.

- Parameter exchange: The SEPPs exchange security related configuration parameters that they need to protect HTTP messages exchanged between the two Network Functions (NF) in their respective networks.

- Error handling: The receiving SEPP sends an error signalling message to the peer SEPP when it detects an error on the N32-f interface.

The following security related configuration parameters may be exchanged between the two SEPPs:

a. Modification policy. A modification policy, as specified in clause 13.2.3.4, indicates which IEs can be modified by an IPX provider of the sending SEPP.

b. Data-type encryption policy. A data-type encryption policy, as specified in 13.2.3.2, indicates which types of data will be encrypted by the sending SEPP.

c. Cipher suites for confidentiality and integrity protection, when application layer security is used to protect HTTP messages between them.

d. N32-f precontext identifier values.N32-f precontext identifier values, as specified in clause 13.2.2.4.1, are used by each SEPP to construct a common N32-f context ID that identifies the set of security related configuration parameters applicable to a protected message received from a SEPP in a different PLMN.

\*\*\*\*\*\*\*\*\*\*\*\* CHANGE 3 (clause 13.2.2 N32-c connection between SEPPs) \*\*\*\*\*\*\*\*\*

13.2.2.2 Procedure for Key agreement and Parameter exchange

1. The two SEPPs shall perform the following cipher suite negotiation to agree on a cipher suite to use for protecting NF service-related signalling over N32-f.

1a. The SEPP which initiated the first N32-c connection shall send a Security Parameter Exchange Request message to the responding SEPP including the initiating SEPP’s supported cipher suites. The cipher suites shall be ordered in initiating SEPP’s priority order. The SEPP shall provide a N32-f precontext ID for the responding SEPP. The precontext IDs are 32-bit random integers, represented as 0-left padded strings of hexadecimal digits.

1b. The responding SEPP shall compare the received cipher suites to its own supported cipher suites and shall select, based on its local policy, a cipher suite, which is supported by both initiating SEPP and responding SEPP.

1c. The responding SEPP shall send a Security Parameter Exchange Response message to the initiating SEPP including the selected cipher suite for protecting the NF service-related signalling over N32. The responding SEPP shall provide a N32-f precontext ID for the initiating SEPP.

1d. The SEPPs shall create the N32-f context ID as follows:

Initiating SEPP's N32-f precontext ID | responding SEPP's N32-f precontext ID

2. The two SEPPs may perform the following exchange of Data-type encryption policies and Modification policies. Both SEPPs shall store protection policies sent by the peer SEPP:

2a. The SEPP which initiated the first N32-c connection shall send a Security Parameter Exchange Request message to the responding SEPP including the initiating SEPP's Data-type encryption policies, as described in clause 13.2.3.2, and Modification policies, as described in clause 13.2.3.4.

2b. The responding SEPP shall store the policies if sent by the initiating SEPP.

2c. The responding SEPP shall send a Security Parameter Negotiation Response message to the initiating SEPP with the responding SEPP’s suite of protection policies.

2d. The initiating SEPP shall store the protection policy information if sent by the responding SEPP.

3. The two SEPPs shall exchange IPX security information lists that contain information on IPX public keys or certificates that are needed to verify IPX modifications at the receiving SEPP.

4. The two SEPPs shall export keying material from the TLS session established between them using the TLS export function. For TLS 1.2, the exporter specified in RFC 5705 [61] shall be used. For TLS 1.3, the exporter described in section 7.5 of RFC 8446 [60] shall be used. The exported key shall be used as the master key to derive session keys and IVs for the N32-f context as specified in clause 13.2.4.4.1.

5. The responding SEPP in the first N32-c connection shall now setup a second N32-c connection by establishing a mutually authenticated TLS connection with the peer SEPP.

6. The two SEPPs start exchanging NF to NF service-related signalling over N32-f and may keep the TLS session open for:

- any further N32-c communication that may occur over time while application layer security is applied to N32-f, or

- any further N32-c and N32-f communication, if TLS is used to protect N32-f.

\*\*\*\*\*\*\*\*\*\*\*\* CHANGE 4 (clause 13.2.2 N32-c connection between SEPPs) \*\*\*\*\*\*\*\*\*

13.2.2.3 Procedure for error detection and handling in SEPP

Errors can occur on an active N32-c connection or on one or more N32-f connections between two SEPPs.

When an error is detected, the SEPP shall map the error to an appropriate cause code. The SEPP shall create a signalling message to inform the peer SEPP, with cause code as one of its parameters.

The SEPP shall use the N32-c connection it initiated to send the signalling message to the peer SEPP. If the old N32-c connection has been terminated, it uses a new N32-c connection instead.

If the error occurred in the processing of the one or more N32-f message(s), the SEPP shall include the corresponding message ID (s), obtained from the metadata section of the N32-f message, as a parameter in the signalling message. This allows the peer SEPP to identify the source message(s) (HTTP Request or Response) on which the other SEPP found the error.

NOTE: Local action taken by either SEPP is out of 3GPP scope.

\*\*\*\*\*\*\*\*\*\*\*\* CHANGE 5 (clause 13.2.2.4 N32-f Context) \*\*\*\*\*\*\*\*\*

13.2.2.4.1 N32-f context ID

The N32-f context ID is used to refer to an N32-f context. The SEPPs shall create the N32-f context ID during the N32-c negotiation and use it over N32-f to inform the receiving peer which security context to use for decryption of a received message.

The SEPPs shall create the N32-f context ID by combining the two N32-f precontext IDs, obtained during the N32-c negotiation. To avoid collision of the N32-f context ID value, the SEPPs shall select the N32-f precontext ID as a random value during the exchange over N32-c.

During transfer of application data over N32-f, the SEPP shall include the N32-f context ID in a separate IE in the metadata part of the JSON structure, see clause 13.2.4.2. The receiving SEPP shall use this information to apply the correct key and parameters during decryption and validation.

\*\*\*\*\*\*\*\*\*\*\*\* CHANGE 6 (clause 13.2.4.4 Protection using JSON Web Encryption (JWE))

13.2.4.4.1 N32-f key hierarchy

The N32-f key hierarchy is based on the N32-f master key generated during the N32-c initial handshake by TLS key export. The N32-f key hierarchy consists of two pairs of session keys and two pairs of IV salts, which are used in two different HTTP/2 sessions. In one Session the N32-c initiator acts as the HTTP client and in the second the N32-c responder acts as the client.

If the exported master secret is reused to set up multiple HTTP sessions or to set up new HTTP sessions on stream ID exhaustion, a new, unique, N32-f Context ID shall be generated to avoid key and IV re-use.

The master key shall be obtained from the TLS exporter. The export function takes 3 arguments: Label, Context, Length (in octets) of the desired output. For the N32 Master key derivation, the label shall be "EXPORTER\_3GPP\_N32\_MASTER", the Context shall be "" (the empty string) and the Length shall be 64.

Editor’s Note: The exporter label for this usage should be registered with IANA

The N32 key derivation function N32-KDF shall be based on HKDF [62] and shall use only the HKDF-Expand function as the initial key material has been generated securely:

N32-KDF (label, L) = HKDF-Expand (N32-f master key, "N32" || N32-Context-ID || label, L),

where

- label is a string used for key separation,

- L is the length of output keying material in octets.

Each run of N32-KDF (label, L) produces either one session key or one IV salt.

There are two pairs of session keys and IV salts to be derived.

NOTE: In AES-GCM re-use of one IV may reveal the integrity key (Joux’s Forbidden attack). The binding of session keys and IV salts to N32-f context IDs and labels is essential to protect against inadvertent use of the same key with a repeated IV.

The labels for the JWE keys are:

- "parallel\_request\_key"

- "parallel\_response\_key"

- "reverse\_request\_key", and

- "reverse\_response\_key".

The keys derived with labels starting parallel shall be used for request/responses in an HTTP session with the N32-c initiating SEPP acting as the client (i.e. in parallel to the N32-c connection). The keys derived with the labels starting reverse shall be used for an HTTP session with the N32-c responding SEPP acting as the client.

To generate the IV salts, the length is 8 and the labels are:

- "parallel\_request\_iv\_salt",

- "parallel\_response\_iv\_salt",

- "reverse\_request\_iv\_salt", and

- "reverse\_response\_iv\_salt".

The 96-bit nonce for AES\_GCM shall be constructed as the concatenation of the IV salt (8 octets, 64-bits) and the sequence counter, SEQ, following section 8.2.1 of NIST Special Publication 800-38D [63] :

Nonce = IV salt || SEQ.

The sequence counter shall be a 32-bit unsigned integer that starts at zero and is incremented for each invocation of the encryption. A different sequence counter shall be maintained for each IV salt.

\*\*\*\*\*\*\*\*\*\*\*\* END OF CHANGES