**3GPP TSG-SA3 Meeting #100e *S3-202037-r3***

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

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.0* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **33.535** | **CR** | **0032** | **rev** | **-** | **Current version:** | **16.0.0** |  |
|  | | | | | | | | |
| *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 | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | AKMA SBA interface clarifications | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Ericsson, ZTE, China Mobile, Samsung | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | AKMA | | | | |  | ***Date:*** | | | 2020-08-07 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***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:*** | | The Service Based Interfaces of the procedures need alignment with respect to the figures, the procedure steps and the SBI definitions.Moreover it is a good practise to follow the SBI definitions as in other specifications e.g. summarizing the services provided by an NF in a table and then providing the details of the service definitions. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Propose clarifications on the SBA related clauses. The clarifications are the following:  - The AF does not provide any AKMA services, it only uses AKMA services, therefore any reference to Naf in the specification and figures is removed.  - The NOTE that the AAnF could be co-located with other NFs such as AUSF or NEFis removed since in the context of the SBA architecture, two NFs could always be co-located according to operator deployment scenarios.  - The AUSF does not provide any AKMA related services therefore the Nausf interface references and clauses are removed.  - The UDM provides SBA services that AKMA procedures use but the UDM services are already defined in TS 33.501. However for the sake of completeness a new clause is added.  - The services provided by the AAnF are summarized in a table  - The definition of the Naanf\_AKMA\_KeyRegistration service operation is aligned with the specification.  - The Naanf\_AKMA\_KeyRegistration has changed name to Naanf\_AKMA\_AnchorKey\_Register  - The Naanf\_AKMA\_AFKey service operation is added since its definition is missing from the existing specification. The name of the service operation is changed to Naanf\_AKMA\_ApplicationKey\_Get.  - The services provided by the NEF are summarized in a table  - The definition of the Nnef\_AKMA\_AFKey service operation is aligned with the normative text and the name is changed to Nnef\_AKMA\_ApplicationKey\_Get.  - Affected figures and specification in clauses 6.1 and 6.2 and are changed to reflect the change in the SBI names. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Unclear specification. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 4.1, 4.3, 6.1, 6.2, 7.1.1, 7.1.2, 7.1.X (new clause), 7.2, 7.3.1, 7.3.2, 7.X (new clause) | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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:*** | |  | | | | | | | | |

\*\*\* 1st CHANGE\*\*\*

## 4.1 Reference model

Figure 4.1-1 shows a fundamental network model of AKMA, as well as the interfaces between them.



Figure 4.1-1: Fundamental Network Model for AKMA

NOTE: Figure 4.1-1 shows the case where AAnF is deployed as a standalone function. Deployments can choose to collocate AAnF with AUSF or with NEF according to operators' deployment scenarios.

The AKMA service requires a new logical entity: AKMA Anchor Function (AAnF).

AAnF is the anchor function in the HPLMN that generates the key material to be used between the UE and the AF and maintains UE AKMA contexts.

\*\*\* 2nd CHANGE\*\*\*

## 4.3 AKMA Service Based Interfaces

The following interfaces are involved in AKMA network architecture:

**- Nnef:** Service-based interface exhibited by NEF.

**- Nudm:** Service-based interface exhibited by UDM.

NOTE X: UDM services related to AKMA service are defined in TS 33.501 [2] clause 14.2.2.

**- Naanf:** Service-based interface exhibited by AAnF.

The AAnF interacts with the AUSF and the AF using Service-Based Interfaces. When the AF is located in the operator's network, the AAnF shall use Service-Based Interface to communicate with the AF directly. When the AF is located outside the operator's network, the NEF shall be used to exchange the messages between the AF and the AAnF.

\*\*\* 2A CHANGE\*\*\*

## 6.1 Deriving AKMA key after primary authentication

There is no separate authentication of the UE to support AKMA functionality. Instead, it reuses the 5G primary authentication procedure executed e.g. during the UE Registration to authenticate the UE. A successful 5G primary authentication results in KAUSF being stored at the AUSF and the UE.



Figure 6.1-1: Deriving AKMA root key after primary authentication

During the primary authentication procedure, the AUSF interacts with the UDM in order to fetch authentication information such as subscription credentials (e.g. AKA Authentication vectors) and the authentication method using the Nudm\_UEAuthentication\_Get Request service operation. In the response, the UDM may also indicate to the AUSF whether AKMA keys need to be generated for the UE. If the AUSF receives the AKMA indication from the UDM, the AUSF shall store the KAUSF and generate the AKMA Anchor Key (KAKMA) and the A-KID from KAUSF after the primary authentication procedure is successfully completed.

After AKMA key material is generated, the AUSF shall send the generated A-KID, and KAKMA to the AAnF together with the UE SUPI using the Naanf\_AKMA\_AnchorKey\_Register Request service operation. The AAnF shall store the latest information sent by the AUSF.

NOTE 1: The AUSF need not store any AKMA key material after delivery to the AAnF.

The UE shall generate the AKMA Anchor Key (KAKMA) and the A-KID from the KAUSF before initiating communication with an AKMA Application Function.

A-KID identifies the KAKMA key of the UE from which other AKMA keys are derived.

A-KID shall be in NAI format as specified in clause 2.2 of IETF RFC 7542, i.e. username@realm. The username part includes the Routing Identifier and the A-TID (AKMA Temporary UE Identifier), and the realm part shall include Home Network Identifier.

The A-TID shall be derived from KAUSF as defined in clause A.3.

NOTE 2: The chance of A-TID collision is not zero but practically low as the A-TID derivation is based on KDF specified in Annex B of TS 33.220 [4]. The detection of A-TID collision as well as potential handling of collision is not addressed in the present document.

The key derivation of KAKMA shall be performed using the key derivation function (KDF) specified in TS 33.220 [4]. KAKMA is computed (as per Annex A.2) as KAKMA=KDF (KAUSF, "AKMA", SUPI), where the key derivation parameters consist of a static string "AKMA", and SUPI.

Since AKMA keys are based on KAUSF from primary authentication run, the AKMA keys can only be refreshed by running a fresh primary authentication.

\*\*\* 2B CHANGE\*\*\*

## 6.2 Deriving AKMA Application Key for a specific AF

Figure 6.2-1 shows the procedure used by the AF to request application function specific AKMA keys from 5GC directly, when the AF is located in the operator's network.



Figure 6.2-1: KAF generation from KAKMA

Before communication between the UE and the AKMA AF can start, the UE and the AKMA AF needs to know whether to use AKMA. This knowledge is implicit to the specific application on the UE and the AKMA AF.

1. When the UE initiates communication with the AKMA AF, it shall include the derived A-KID in the Application Session Establishment request message (see clause 6.1).

2. If the AF does not have an active context associated with the A-KID, then the AF sends a Naanf\_AKMA\_ApplicationKey\_Get request to AAnF with the A-KID to request the AKMA Application Key for the UE. The AF also includes its identity (AF Id) in the request. The AAnF shall authorize AF. The AAnF shall check whether the AAnF can provide the service to the AF based on the configured local policy or based on the authorization information or policy provided by the NEF/NRF using the AF Id. If succeeds, the following procedures are executed. Otherwise, the AAnF shall reject the procedure.

The AAnF can check whether the subscriber is authorized to use AKMA by the presence of the AKMA anchor key K\_AKMA that has been received from the AUSF.

If the AAnF is in possession of the AKMA Application Key (KAF), it responds to the AF with the KAF. If not, the AAnF shall check if it has the UE specific KAKMA key identified by the A-KID.

If KAKMA is available in AAnF, the AAnF shall continue with step3.

If KAKMA is not available, the AAnF shall continue with step 4 and send an error response.

3. The AAnF derives the AKMA Application Key (KAF) from KAKMA.

The key derivation of KAF shall be performed using the key derivation function (KDF) specified in TS 33.220 [4]. KAF is computed (as per clause A.4) as KAF=KDF (KAKMA, AF\_ID), where the AF\_ID is constructed as follows: AF\_ID = FQDN of the AF || Ua\* security protocol identifier. The Ua\* security protocol identifier is specified as Ua security protocol identifier in Annex H of TS 33.220 [4]. The key used for the derivation of KAF is KAKMA.

4. The AAnF sends Naanf\_AKMA\_ApplicationKey\_Get response to the AF with KAF and lifetime.

5. The AF response the Application Session Establishment request to the UE.

\*\*\* 3rd CHANGE\*\*\*

## 7.1 Services Provided by AAnF

### 7.1.1 General

The following table shows the AAnF Services and AAnF Service Operations.

Table 7.1.1-1: List of AAnF Services

|  |  |  |  |
| --- | --- | --- | --- |
| Service Name | Service Operations | Operation  Semantics | Example Consumer(s) |
| Naanf\_AKMA | AnchorKey\_Register | Request/Response | AUSF |
| ApplicationKey\_Get | Request/Response | AF, NEF |

### 7.1.2 Naanf\_AKMA\_AnchorKey\_Register service operation

**Service operation name:** Naanf\_AKMA\_AnchorKey\_Register.

**Description:** The NF consumer requests the AAnF to store the AKMA related key material.

**Input, Required:** SUPI, A-KID, KAKMA

**Input, Optional:** None.

**Output, Required:** None.

**Output, Optional:** None.

\*\*\* 4th CHANGE\*\*\*

### 7.1.X Naanf\_AKMA\_ApplicationKey\_Get service operation

**Service operation name:** Naanf\_AKMA\_ApplicationKey\_Get.

**Description:** The NF consumer requests the AAnF to provide AF related key material.

**Input, Required:** A-KID, AF\_ID

**Input, Optional:** None.

**Output, Required:** KAF, KAF expiration time.

**Output, Optional:** None.

\*\*\* 5th CHANGE\*\*\*

## 7.2 Void

### 

\*\*\* 6th CHANGE\*\*\*

## 7.3 Services Provided by NEF

### 7.3.1 General

The NEF exposes AKMA Application Key derivation service to the requester NF.

The following table shows the NEF Services and NEF Service Operations related to AKMA service.

Table 7.1.1-1: List of AAnF Services

|  |  |  |  |
| --- | --- | --- | --- |
| Service Name | Service Operations | Operation  Semantics | Example Consumer(s) |
| Nnef\_AKMA | ApplicationKey\_Get | Request/Response | AF |

### 7.3.2 Nnef\_AKMA\_ApplicationKey\_Get service operation

**Service operation name:** Nnef\_AKMA\_ApplicationKey\_Get.

**Description:** The NF consumer requests the NEF to provide AF related key material.

**Input, Required:** A-KID, AF\_ID

**Input, Optional:** None.

**Output, Required:** KAF, KAF expiration time.

**Output, Optional:** None.

\*\*\* 7th CHANGE\*\*\*

## 7.X Services Provided by UDM

UDM services related to AKMA service are defined in TS 33.501 [2] clause 14.2.2.

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