ETSI TS 133 535 V16.1.0 (2020-11)

5G;
Authentication and Key Management for Applications (AKMA) based on 3GPP credentials in the 5G System (5GS)
(3GPP TS 33.535 version 16.1.0 Release 16)

Reference
RTS/TSGS-0333535vG10

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

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 specifies the security features and mechanisms to support authentication and key management aspects for applications based on subscription credential(s) in 5G system as defined in TS 33.501 [2].

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 33.501: "Security architecture and procedures for 5G system".

[3] 3GPP TS 23.501: "System Architecture for the 5G System".

[4] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)".

[5] 3GPP TS 23.222: "Common API Framework for 3GPP Northbound APIs".

[6] IETF RFC 7542: "The Network Access Identifier".

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 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].

AKMA subscription data: The data in the home operator's network indicating whether or not the subscriber is allowed to use AKMA.

AKMA context: A set of parameters stored in AAnF, including SUPI, K_{AKMA} and A-KID.

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].

A-KID	AKMA Key IDentifier
A-TID	AKMA Temporary UE IDentifier
AAnF	AKMA Anchor Function

AF	Application Function
AKMA	Authentication and Key Management for Applications
AMF	Access and Mobility Management Function
AUSF	AUthentication Server Function
K_{AF}	AKMA Application Key
K_{AKMA}	AKMA Anchor Key
KDF	Key Derivation Function
NEF	Network Exposure Function
UDM	Unified Data Management

4 Architecture for AKMA

4.1 Reference model

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

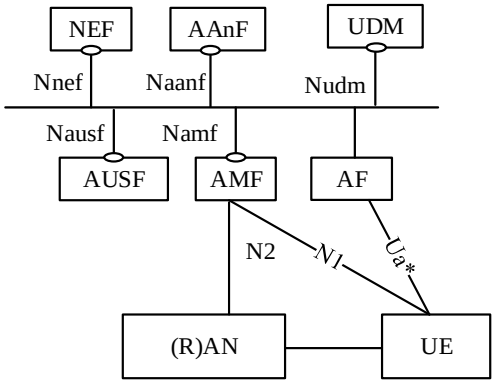

Figure 4.1-1: Fundamental Network Model for AKMA

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, called the AKMA Anchor Function (AAnF).

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

4.2 Network elements

4.2.1 AAnF

AAnF stores the AKMA Anchor Key (K_{AKMA}) for AKMA service, which is received from the AUSF after the UE completes a successful 5G primary authentication.

4.2.2 AF

The AF is defined in TS 23.501 [3] with additional functions:

- AF with the AKMA service enabling requests for AKMA Application Key, called $K_{AF,}$ from the AAnF using A-KID.
- AF shall be authenticated and authorized by the operator network before providing the K_{AF} to the AF.

4.2.3 NEF

The NEF is defined in TS 23.501 [3] with additional functions:

- The NEF enables and authorizes the external AF assessing AKMA service and forwards the request towards the AAnF.
- The NEF performs the AAnF selection.

4.2.4 AUSF

The AUSF is defined in TS 23.501 [3] with additional functions:

- AUSF provides the SUPI and AKMA key material (A-KID,K_{AKMA}) of the UE to the AAnF.

4.2.5 UDM

The UDM is defined in TS 23.501 [3] with the additional functions:

- UDM stores AKMA subscription data of the subscriber.

4.3 AKMA Service Based Interfaces(SBIs)

4.3.0 General

The following interfaces are involved in AKMA network architecture:

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

NOTE 1: 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.

4.3.1 Reference point Ua*

The reference point Ua* carries the application protocol, which is secured using the key material agreed between UE and AAnF as a result of successful AKMA procedures.

4.4 Security requirements and principles for AKMA

4.4.0 General

The following security requirements are applicable to AKMA:

- AKMA shall reuse the same UE subscription and the same credentials used for 5G access.
- AKMA shall reuse the 5G primary authentication procedure and methods specified in TS 33.501 [2] for the sake of implicit authentication for AKMA services.
- The SBA interface between the AAnF and the AUSF shall be confidentiality, integrity and replay protected.
- The SBA interface between AAnF and AF/NEF shall be confidentiality, integrity and replay protected.

- The AKMA Application Key (K_{AF}) shall be provided with a maximum lifetime.

NOTE: Roaming aspects are not considered in the present document.

4.4.1 Requirements on Ua* reference point

The Ua* reference point is application specific. The generic requirements for Ua* are:

- Ua* protocol shall be able to carry AKMA Key Identifier (A-KID);
- the UE and the AKMA AF shall be able to secure the reference point Ua* using the AKMA Application Key derived from the AKMA Anchor Key.

NOTE 1: The exact method of securing the reference point Ua* depends on the application protocol used over reference point Ua*.

NOTE 2: Specifying Ua* protocol identifier is not considered in the present document.

- The Ua* protocol shall be able to handle the expiration of $K_{AF.}$

4.4.2 Requirements on AKMA Key Identifier (A-KID)

Requirements for AKMA Key Identifier (A-KID) are:

- A-KID shall be globally unique;
- A-KID shall be usable as a key identifier in protocols used in the reference point Ua*;
- AKMA AF shall be able to identify the AAnF serving the UE from the A-KID.

5 Key management

5.1 AKMA key hierarchy

The key hierarchy (see Figure 5.1-1) includes the following keys: K_{AUSF}, K_{AKMA}, $K_{AF.}$ K_{AUSF} is generated by AUSF as specified in clause 6 of TS 33.501 [2].

Keys for AAnF:

- K_{AKMA} is a key derived by ME and AUSF from K_{AUSF}.

Keys for AF:

- K_{AF} is a key derived by ME and AAnF from K_{AKMA}.

K_{AKMA} and K_{AF} are derived according to the procedures of clauses 6.1 and 6.2.

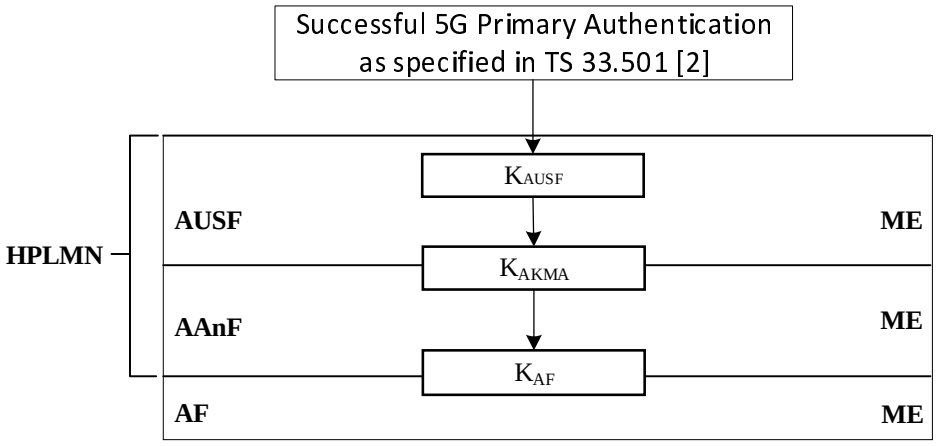

Figure 5.1-1: AKMA Key Hierarchy

5.2 AKMA key lifetimes

The K_{AKMA} and A-KID are valid until the next successful primary authentication is performed (implicit lifetime), in which case the K_{AKMA} and A-KID are replaced.

AKMA Application Keys K_{AF} shall use explicit lifetimes based on the operator's policy. The lifetime of K_{AF} shall be sent by the AAnF as described in clauses 6.2 and 6.3. In case that a new AKMA Anchor Key K_{AKMA} is established, the AKMA Application Key K_{AF} can continue to be used until its lifetime expires. When the K_{AF} lifetime expires, a new AKMA Application Key is established based on the current AKMA Anchor Key K_{AKMA}.

6 AKMA Procedures

6.1 Deriving AKMA key after primary authentication

There is no separate authentication of the UE to support AKMA functionality. Instead, AKMA reuses the 5G primary authentication procedure executed e.g. during the UE Registration to authenticate the UE. A successful 5G primary authentication results in K_{AUSF} being stored at the AUSF and the UE. Figure 6.1-1 shows the procedure to derive K_{AKMA} after a successful primary authentication.

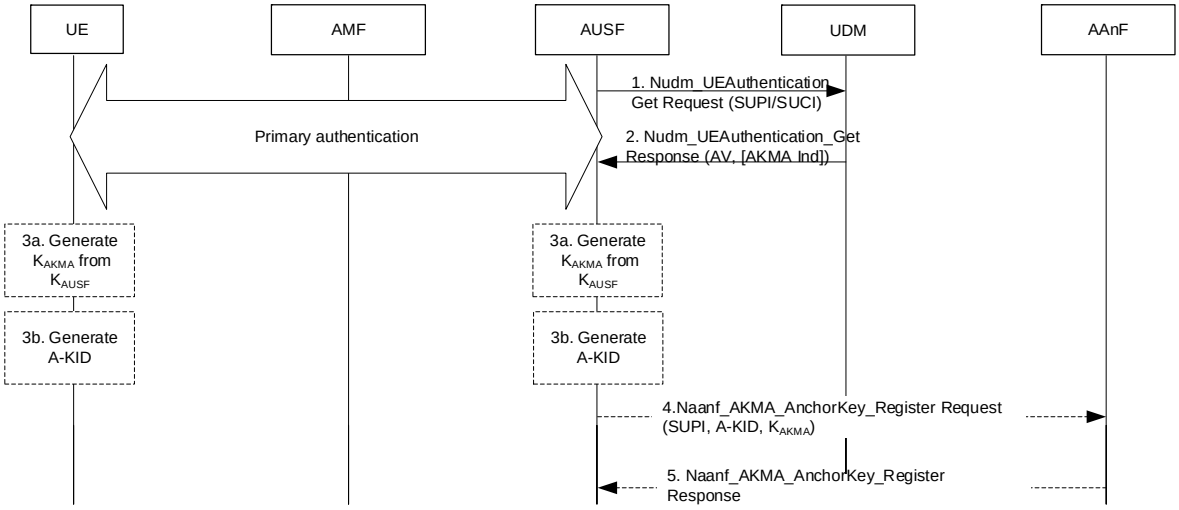

Figure 6.1-1: Deriving K_{AKMA} after primary authentication

1) 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.

2) In the response, the UDM may also indicate to the AUSF whether AKMA keys need to be generated for the UE.

3) If the AUSF receives the AKMA indication from the UDM, the AUSF shall store the K_{AUSF} and generate the AKMA Anchor Key (K_{AKMA}) and the A-KID from K_{AUSF} after the primary authentication procedure is successfully completed.

   The UE shall generate the AKMA Anchor Key (K_{AKMA}) and the A-KID from the K_{AUSF} before initiating communication with an AKMA Application Function.

4) After AKMA key material is generated, the AUSF shall send the generated A-KID, and K_{AKMA} to the AAnF together with the SUPI of the UE using the Naanf_AKMA_KeyRegistration 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.

NOTE 1a: When re-authentication runs, the AUSF generates a new A-KID, and a new KAKMA and sends the new generated A-KID and KAKMA to the AAnF. After receiving the new generated A-KID and KAKMA, the AAnF deletes the old A-KID and KAKMA and stores the new generated A-KID and KAKMA.

5) The AAnF sends the response to the AUSF using the Naanf_AKMA_AnchorKey_Register Response service operation.

A-KID identifies the K_{AKMA} key of the UE.

A-KID shall be in NAI format as specified in clause 2.2 of IETF RFC 7542 [6], i.e. username@realm. The username part shall include 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 K_{AUSF} as specified in Annex 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.

K_{AKMA} shall be derived from K_{AUSF} as specified in Annex A.2. Since AKMA keys are derived from K_{AUSF} based on primary authentication run, the AKMA keys can only be refreshed by a new successful primary authentication.

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 the AAnF, when the AF is located inside the operator's network.

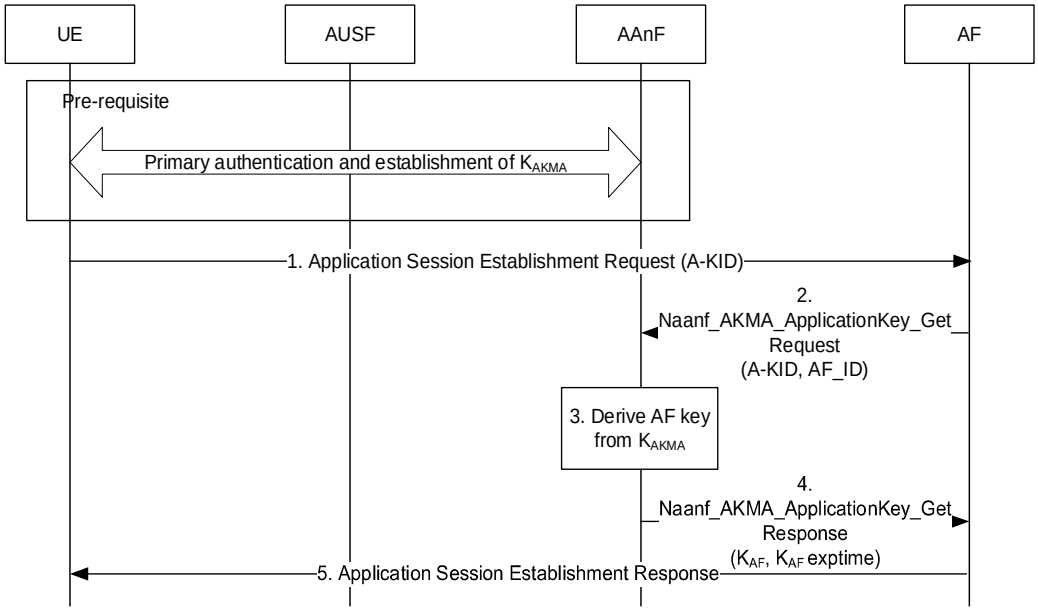

Figure 6.2-1: K_{AF} generation from K_{AKMA}

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 or indicated by the AKMA AF to the UE (see clause 6.5).

1. The UE shall generate the AKMA Anchor Key (K_{AKMA}) and the A-KID from the K_{AUSF} before initiating communication with an AKMA Application Function. When the UE initiates communication with the AKMA AF, it shall include the derived A-KID (see clause 6.1) in the Application Session Establishment request message.

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 K_{AF} for the UE. The AF also includes its identity (AF ID) in the request.

   AF ID consists of the FQDN of the AF and the Ua* security protocol identifier. The latter parameter identifies the security protocol that the AF will use with the UE.

   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 NRF using the AF ID. If it succeeds, the following procedures are executed. Otherwise, the AAnF shall reject the procedure.

   The AAnF shall verify whether the subscriber is authorized to use AKMA based on the presence of the UE specific K_{AKMA} key identified by the A-KID.

      If K_{AKMA} is present in AAnF, the AAnF shall continue with step 3.

      If K_{AKMA} is not present in the AAnF, the AAnF shall continue with step 4 with an error response.

3. The AAnF derives the AKMA Application Key (K_{AF}) from K_{AKMA} if it does not already have K_{AF}.

   The key derivation of K_{AF} shall be performed as specified in Annex A.4.

4. The AAnF sends Naanf_AKMA_ApplicationKey_Get response to the AF with K_{AF} and the K_{AF} expiration time.

5. The AF sends the Application Session Establishment Response to the UE. If the information in step 4 indicates failure of AKMA key request, the AF shall reject the Application Session Establishment by including a failure cause. Afterwards, UE may trigger a new Application Session Establishment request with the latest A-KID to the AKMA AF.

6.3 AKMA Application Key request via NEF

Figure 6.3-1 shows the procedure used by the AF to request K_{AF} from the AAnF via NEF, when the AF is located outside the operator's network.

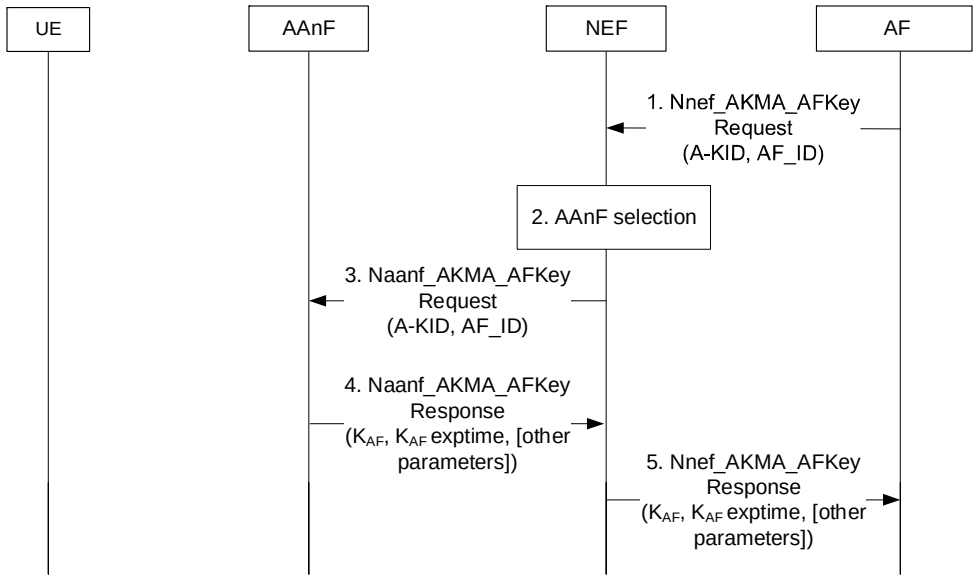

Figure 6.3-1: AKMA Application Key request via NEF

1. When the AF is about to request AKMA Application Key for the UE from the AAnF, e.g. when UE initiates application session establishment request as in clause 6.2, the AF discovers the HPLMN of the UE based on the A-KID and sends the request towards the AAnF via NEF service API. The request shall include the A-KID and the AF ID.

NOTE: In the case of architecture without CAPIF support, the AF is locally configured with the API termination points for the service. In the case of architecture with CAPIF support, the AF obtains the service API information from the CAPIF core function via the Availability of service APIs event notification or Service Discover Response as specified in TS 23.222 [5].

2. If the AF is authorized by the NEF to request K_{AF}, the NEF discovers and selects an AAnF based on local configuration or via NRF in the same way as the AF selects the AAnF in clause 6.2.

3. The NEF forwards the K_{AF} request to the selected AAnF.

4. The AAnF generates the K_{AF} as specified in clause 6.2 and sends the response to the NEF with the K_{AF}, the K_{AF} expiration time (K_{AF}_exptime) and potentially other parameters.

5. The NEF forwards the response to the AF.

Editor's Note: Whether other parameters are to be returned to the AF via NEF is FFS.

6.4 AKMA key change

6.4.1 K_{AKMA} re-keying

K_{AKMA} shall be re-keyed by running a successful primary authentication as described in clause 6.1.

6.4.2 K_{AF} re-keying

The K_{AF} re-keying depends on the lifetime of the K_{AF} and may be trigged by the AF, which means that when a new K_{AKMA} is derived, the K_{AF} will not be re-keyed automatically.

When the lifetime of K_{AF} expires, the AF may reject UE's access to the AF based on its policy. If there has been a change of K_{AKMA} (e.g., due to a successful run of primary authentication), the UE may re-try accessing the AF by using the A-KID derived from the new K_{AKMA}.

6.4.3 K_{AF} refresh

Ua* protocol may support refresh of K_{AF}. If the Ua* protocol supports refresh of K_{AF}, the AF may refresh the K_{AF} at any time using the Ua* protocol.

6.5 Initiation of AKMA

In case when the UE does not know to use AKMA for a service, then the following procedure shown in figure 6.5-1 applies.

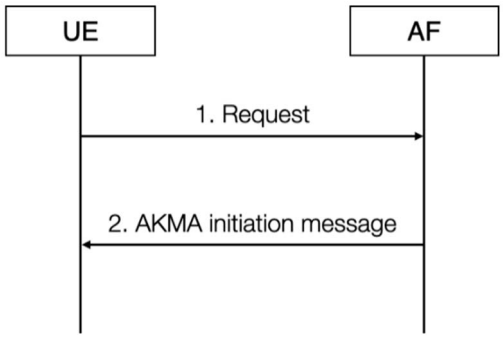

Figure 6.5-1: Initiation of AKMA

1. The UE may start communication over reference point Ua* with the AF with or without any AKMA-related parameters.
2. If the AF requires the use of shared keys obtained by means of the AKMA, but the request from UE does not include AKMA-related parameters, the AF replies with an AKMA initiation message. The form of this initiation message may depend on the particular reference point Ua*.

In case the UE knows to use AKMA for a service, then it directly initiates the procedure in clause 6.2.

7 Security related services

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, K_{AKMA}

Input, Optional: None.

Output, Required: None.

Output, Optional: None.

7.2 Void

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_Getservice 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: K_{AF}, K_{AF} expiration time.

Output, Optional: None.

7.4 Services provided by UDM

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

Annex A (normative): Key derivation functions

A.1 KDF interface and input parameter construction

A.1.1 General

All key derivations for AKMA shall be performed using the key derivation function (KDF) specified in Annex B.2.2 of TS 33.220 [4].

This clause specifies how to construct the input string, S, and the input key, KEY, for each distinct use of the KDF. Note that "KEY" is denoted "Key" in TS 33.220 [4].

A.1.2 FC value allocations

The FC number space used is controlled by TS 33.220 [4], FC values allocated for the present document are in the range of 0x80 – 0x82.

A.2 K_{AKMA} derivation function

When deriving a K_{AKMA} from K_{AUSF}, the following parameters shall be used to form the input S to the KDF:

- FC = 0x80;
- P0 = "AKMA";
- L0 = length of "AKMA"; (i.e. 0x00 0x04)
- P1 = SUPI;
- L1 = length of SUPI.

The input key KEY shall be K_{AUSF}.

SUPI shall be have the same value as parameter P0 in Annex A.7.0 of TS 33.501 [2].

A.3 A-TID derivation function

When deriving the A-TID from K_{AUSF}, the following parameters shall be used to form the input S to the KDF:

- FC = 0x81;
- P0 = "A-TID";
- L0 = length of "A-TID"; (i.e. 0x00 0x05)
- P1 = SUPI;
- L1 = length of SUPI.

The input key KEY shall be K_{AUSF}.

SUPI shall be have the same value as parameter P0 in Annex A.7.0 of TS 33.501 [2].

A.4 K_{AF} derivation function

When deriving a K_{AF} from K_{AKMA}, the following parameters shall be used to form the input S to the KDF:

- FC = 0x82;
- P0 =AF_ID;
- L0 = length of AF_ID

The input key KEY shall be K_{AKMA}.

AF_ID is constructed as follows:

AF_ID = FQDN of the AF || Ua* security protocol identifier, where the Ua* security protocol identifier is specified as Ua security protocol identifier in Annex H of TS 33.220 [4].

Annex B (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2020-06	SA#88-e	SP-200381				EditHelp review. Presented for information and approval	1.0.0
2020-07	SA#88-e					Upgrade to change control version	16.0.0
2020-09	SA#89-e	SP-200708	0001	-	D	Add Abbreviations to clause 3.3	16.1.0
2020-09	SA#89-e	SP-200708	0009	1	F	Clarifications on error response handling in AKMA process	16.1.0
2020-09	SA#89-e	SP-200708	0013	1	F	Re-authentication in AKMA	16.1.0
2020-09	SA#89-e	SP-200708	0020	-	F	Adding AKMA context description	16.1.0
2020-09	SA#89-e	SP-200708	0023	1	F	Corrections and clarifications to clause 4	16.1.0
2020-09	SA#89-e	SP-200708	0024	1	F	Corrections to AKMA key lifetimes	16.1.0
2020-09	SA#89-e	SP-200708	0025	1	F	Corrections and clarifications to AKMA procedures	16.1.0
2020-09	SA#89-e	SP-200708	0026	1	F	Assignment of FC values for key derivations	16.1.0
2020-09	SA#89-e	SP-200708	0027	-	F	Specification of value of SUPI for key derivations	16.1.0
2020-09	SA#89-e	SP-200708	0032	1	F	AKMA SBA interface clarifications	16.1.0
2020-09	SA#89-e	SP-200708	0034	1	F	Several clarifications and editorials	16.1.0

History

Document history		
V16.0.0	July 2020	Publication
V16.1.0	November 2020	Publication