3GPP TSG SA WG3 Security — SA3#35 **S3-040741**
October 5-8, 2004
St Paul's Bay, Malta

Source: Nokia, Siemens, Huawei

Title: GBA User Security Settings (GUSS) usage

Agenda item: 6.9.2 (GBA)

Document for: Information/Decision

1 Introduction

At SA3#34 is was decided that GBA User Security Settings (GUSS) will be used in transferring identity and authorization information from the HSS to the BSF, and application specific User Security Setting (USS) will be used transfer the identity and authorization information from the BSF to the NAF (see S3-040650). This discussion paper discusses the content of these information elements as they have been defined in CN4 (sections 2.1 and 2.2), describes the procedrue related to the GUSS usage (section 2.3), and lists some open questions with tentative answers related to the GUSS usage (section 2.4).

2 Discussion

2.1 GBA User Security Setting (USS)

S3-040650 defined GBA User Security Setting (USS) the following way:

> *"An application-specific parameter set describing the security related usage of bootstrapping function by the BSF and, optionally, some types of NAFs in the context of an application and in relation to a subscriber. A user security setting has two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g IMPI, IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g. access to application allowed, type of certificates which may be issued). Sometimes also called application-specific user security setting."*

An example of a USS that is based on the XML Schema defined by CN4 (see annex A) is given below.

CN4 has defined an XML Schema (N4-041005) that is also listed in Annex A. Below is an example of an USS.

```
<uss id="1234567890" type="1">
    <uids>
        <uid>tel:358504837438</uid>
        <uid>pekka.laitinen@nokia.com</uid>
        ...
    </uids>
    <flags>
        <flag>1</flag>
        ...
    </flags>
</uss>
```

The <uss> element contains list of user identifiers inside <uids> element and authorization flags inside <flags> element. A <uss> element can be identified by the "id" attribute that contains the application identifier. The element also contains "type" attribute that indicates the type of the application, e.g., PKI portal (1), authentication proxy (2), or Presence admin server (3). The meaning of separate authorization flags are directly liked to the application type, e.g., for the PKI portal type flag "1" would mean that the issuance of authentication certificate is allowed, and "2" would meant the non-repudiation certificate issuance is allowed for the subscriber. If the particular flags are missing then the corresponding action is not allowed.

The NAF can request one or more <uss> elements from the BSF by listing the application identifiers in the request to the BSF. The BSF will locate the corresponding <uss> elements and return them to the NAF if it is authorized to receive them. Whether a NAF is authorized to receive a particular <uss> element is a configuration option in the BSF.

2.2 GBA User Security Settings (GUSS)

S3-040650 defined GBA User Security Settings (GUSS):

> "The set of all application-specific user security settings."

An example of a GUSS that is based on the XML Schema defined by CN4 is given below. For extension of this definition proposed by this document see annex A.

```
<guss id="358500004837438@ims.mnc050.mcc358.3gppnetwork.org">
    <ussList>
        <uss id="1234567890" type="1">
            <uids>
                <uid>tel:358504837438</uid>
                <uid>pekka.laitinen@nokia.com</uid>
                ...
            </uids>
            <flags>
                <flag>1</flag>
                ...
            </flags>
        </uss>
        ...
    </ussList>
</guss>
```

The <guss> element contains a list of <uss> elements for the particular subscriber. A <guss> can be identified by the "id" attribute that contains the IMPI (or pseudo-IMPI derived from the IMSI) of the subscriber.

2.3 GUSS/USS procedure

2.3.1 Successful case

During the bootstrapping procedure the BSF fetches subscriber's GUSS from the HSS and stores it along with other bootstrapping information such as the B-TID, key material, and key lifetime values. When the UE contacts the NAF, the NAF will send a request to the BSF containing:

1. B-TID (received from the UE),
2. NAF hostname (that the UE used to contact the NAF), and
3. zero or more application-ids (each identifying an application/service).

Upon receiving this information, the BSF does the following:

4. verifies that the NAF is authorized to use the hostname given,
5. locates the bootstrapping parameters identified by the B-TID,
6. derives the NAF specific key(s) using the given hostname,
7. locates USSs identified by the application-ids given in the request,
8. verifies that the NAF is authorized to receive each of these USSs, and
9. checks whether the existence of certain USSs is required for the NAF.

If all this is successful, the BSF returns the following parameters to the NAF:

10. key lifetime,

11. NAF specific key(s), and
12. zero or more requested USSs.

Upon receiving this information, the NAF does the following:

13. concludes the authentication procedure with the UE, and
14. examines USSs for relavant information.

The USS may contain authorization flag(s) to instruct the NAF to either allow or disallow access to the service. This is done for example in the PKI portal case, where the USS will contain information whether the PKI portal is allowed to issue a certain kind of certificate.

2.3.2 (Potential) error cases

Related to subscriber's GUSS or USSs, there are three potential error cases possible:

1. The subscriber does not have a GUSS in the HSS,
2. The subscriber does not have a particular USS that is requested by the NAF, and
3. The NAF is not authorized to access the requested USS.

The first case marks the situation where the subscriber does not have any GBA related data, i.e., no GUSS in the HSS. As GBA should be usable to all subscribers this should not result to an error. But is should be noted that without of any USSs in the HSS, NAFs will not receive subscriber's identities or authorization flags. The NAF obtains only the information that the UE is a valid subscriber of the operator with an exception that special NAFs (according to local BSF policy set by the operator) may receive the private identity as well.

The second case may be caused by several reasons:

a) Subscriber does not have a subscription of the service;

b) Subscriber has a subscription of the service but for some reason is not allowed to use the service.

In general case, a single NAF may host several services each requiring a separate USS, e.g., an authentication proxy, and a subscriber accessing the NAF may have subscribed only to one of those services, the subscriber will have only one USS that the NAF is requesting from the BSF. Since the NAF may not know which service the subscriber is going to access during the authentication phase, it must request USSs for all the services that it is hosting. If this case generates an error message from the BSF to the NAF, the subscriber would not be able to access the service. Therefore a missing USS should not cause an error message in general, and it should be the NAF that handles the missing USS scenario according to its internal policies.

In specific cases, e.g., with visited NAFs, more stricter access control may be needed. In this case, it would be required that the NAF must request one or more USSs and that those USSs must exist for the subscriber. If even one USS is missing from subscriber's GUSS, it would cause an error message being sent to the NAF. This can be implemented in the BSF as a local policy, and it would be enforced in step 9 in section 2.3.1.

So, in the generic case a missing USS would not result to an error, but in the specific cases it would.

Note: According to the current TS 29.109, if the NAF requested one or more USSs by giving one or multiple application ids in the request over Zn interface, and if any of the USS are missing from subscriber's GUSS, the BSF shall return error: USS not found. This needs to be changed in TS 29.109.

The third case should result to an error message as the NAF is trying to request a USS that it is not authorized to use.

2.4 Questions

1) Should visited NAFs receive USSs?

The particular USS that a visited NAF requests from the BSF is already subjected to access control, i.e., the BSF knows which NAFs are allowed to receive which USSs. Hence, a visited NAF requesting a particular USS should be allowed to receive the USS if the access control check by the BSF is successful.

Tentantative answer: yes.

2) Should the BSF be able to check the content of an USS?

Since the meaning of th eauthorization flag values depend on the corresponding application types, and new application types can be defined which are not standardized, the BSF should not be required to check the content of USSs.

Tentative answer: no.

3) If a service has been temporarily revoked for a subscriber how this should be indicated in GAA?

Two ways have been identified to handle this:

1. The HSS temporarily removes the corresponding <uss> elelement from subscriber's <guss> element that is sent to the BSF.

2. The HSS uses a "status" parameter in the <uss> element to indicate the current status of the service for the particular subscriber (e.g., <uss id="123..." type="1" status="non-active"/>)

In both cases, the HSS would change the "active" <guss> element in the HSS when subscriber's access to the particular service is revoked.

With option 1, the BSF does not find the <uss> element that the NAF requested. The NAF will receive error message with APPLICATION_ID_UNKNOWN. The NAF will assume that the subscriber does not have a subscription and will not allow the subscriber to access the service.

With option 2, the BSF does find the <uss> element but by checking the "status" parameter it finds out that the service has been temporarily removed, and will not send the <uss> element to the NAF. In NAFs point of view, the difference between the two options are that with option 2 it possible to indicate to the NAF that the subscription exists but it has been temporarily revoked. It should be noted that the HSS should be responsible to set the "status" parameter accordingly in the <uss> element.

Note: The "status" parameter has not been specified in TS 29.109 as it is not certain this is needed.

Tentantive answer: Option 1 seems to be sufficient as the "non-active" or alike USSs do not need to be transferred to the BSF. Also, it is unclear whether the NAF needs to know the status. Bottom line is that the subscriber is not allowed to access the service.

4) How is information intended for the BSF (such as the GBA_U indication) sent from the HSS over Zh?

There are several ways to send the GBA_U indication (and other possible parameters intended for the BSF) over Zh reference point:

1. Add an AVP for each parameter.

2. Use GUSS to send the parameters.

Option 1 seems to be sufficient if the GBA_U indication is the only parameter being sent from the HSS to the BSF. Howver, if there is need to transfer also other parameters (e.g., subscriber specific key lifetime) then option 2 is better as it can be extended easier and there is no need to add new AVPs for each attributes that needs to be transferred from the HSS to the BSF since only the AVP transferring the <guss> element would be needed.

Tentative answer: Option 2 seems better alternative. An BSF specific information element can be added inside the <guss> element called, e.g., <bsfInfo> which can contain the needed parameters for the BSF. If the <bsfInfo> element is not available for a subscriber (i.e., either the <guss> or <bsfInfo> element does not exist for the subscriber), then the BSF will use the default values in the BSF local policy defined by the MNO.

3 Proposal

We ask SA3 to endorse the GUSS/USS procedure descriptions in section 2.3 and the tentative answers in section 2.4. Attached CR implements the required changes to TS 33.220.

Annex A: XML Schema for GUSS

CN4 defined the XML Schema for the GUSS in N4-041164:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="guss-schema-of-3gpp-gaa"
    xmlns:tns="guss-schema-of-3gpp-gaa"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

  <!-- This import brings in the XML language attribute xml:lang-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
      schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <!—- The whole user’s GBA specific data set  -->
  <xs:complexType name="guss">
    <xs:sequence>
       <xs:element ref="bsfInfo" minOccurs="0"/>
       <xs:element ref="ussList"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
  </xs:complexType>

  <!—- BSF specific information element -->
  <xs:complexType name="bsfInfo">
    <xs:sequence>
       <xs:element name="uiccType" type="xs:string" minOccurs="0"/>
       <xs:element name="lifetime" type="xs:integer" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <!—-List of all users individual User Security Settings -->
  <xs:complexType name="ussList">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
       <xs:element ref="uss"/>
    </xs:sequence>
  </xs:complexType>

  <!—- User Security Setting data -->
  <xs:complexType name="uss">
    <xs:sequence>
      <xs:element ref="uids"/>
      <xs:element name="flags"/>
    </xs:sequence>
    <xs:attribute name="id"   use="required" type="xs:string"/>
    <xs:attribute name="type" use="required" type="xs:int"/>
  </xs:complexType>

  <!—- User Public Identities for authentication -->
  <xs:complexType name="uids">
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element name="uid"  type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

  <!—- GAA Application type specific Authorization flag codes -->
  <xs:complexType name="flags">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="flag"  type="xs:int"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

The revision marked part of the XML schema marks the needed changes regarding the tentative answer to the question 4 in section 2.4.

3GPP TSG-SA WG3 Meeting S3#35
St Paul's Bay, Malta, October 5-8, 2004

Tdoc ⌘Att_S3-040741

CR-Form-v7.1

CHANGE REQUEST

⌘	**33.220**	**CR**	**020**	⌘ **rev**	**-**	⌘	Current version:	**6.2.0**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ □ ME □ Radio Access Network □ Core Network ☒

Title:	⌘	GBA User Security Settings (GUSS) usage in GAA		
Source:	⌘	Nokia, Siemens, Huawei		
Work item code:	⌘	SEC1-SC	***Date:*** ⌘	27/09/2004
Category:	⌘	**C**	***Release:*** ⌘	Rel-6
		Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction 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.		*Use one of the following releases:* *Ph2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *Rel-4 (Release 4)* *Rel-5 (Release 5)* *Rel-6 (Release 6)* *Rel-7 (Release 7)*

Reason for change:	⌘	The details of GBA user security settings (GUSS) are used is missing.
Summary of change:	⌘	- The BSF may require that the NAF is required to ask for one or more USSs from the BSF over Zn reference point. If one or more of these USSs are missing from the request, the BSF will not provide bootstrapping information to the NAF. - If a NAF requests USSs from the BSF and they are not present in user GBA user security settings, it will not cause an error. The BSF will send only the requested and found USSs to the NAF. - GUSS may be used to transfer subscriber specific parameters intended for the BSF only (i.e., the type of subscriber's UICC and subscriber specific key lifetime). - The complete set of application-specific user security settings are addressed as GUSS and application-specific user security setting as USS in the specification for clarity reasons.
Consequences if not approved:	⌘	The details of how GBA user security settings (GUSS) are used are missing.

Clauses affected:	⌘	3.1, 3.2, 4.2.1, 4.2.2, 4.4.3, 4.4.6, 4.5.3, 5.3.3				
		Y	**N**			
Other specs affected:	⌘	**X**		Other core specifications	⌘	TS 29.109
			X	Test specifications		
			X	O&M Specifications		
Other comments:	⌘					

===== BEGIN CHANGE =====

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Bootstrapping Server Function: BSF is hosted in a network element under the control of an MNO.

Editor's note: Definition to be completed.

ME-based GBA: in GBA_ME, all GBA-specific functions are carried out in the ME. The UICC is GBA-unaware. If the term GBA is used in this document without any further qualification then always GBA_ME is meant, see clause 4 of this specification.

UICC-based GBA: this is a GBA with UICC-based enhancement. In GBA_U, the GBA-specific functions are split between ME and UICC, see clause 5 of this specification.

Network Application Function: NAF is hosted in a network element under the control of an MNO.

Editor's note: Definition to be completed.

Bootstrapping Transaction Identifier: the bootstrapping transaction identifier (B-TID) is used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.

GBA User Security Setting: USS is ~~A~~an application-specific parameter set describing the security related usage of bootstrapping function by the BSF and, optionally, some types of NAFs in the context of an application and in relation to a subscriber. A user security setting has two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g. IMPI, IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g. access to application allowed, type of certificates which may be issued). Sometimes also called application-specific user security setting.

GBA User Security Settings: GUSS contains the BSF specific information element and the set of all application-specific ~~user security settings~~USSs.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AK	Anonymity Key
AKA	Authentication and Key Agreement
B-TID	Bootstrapping Transaction Identifier
BSF	Bootstrapping Server Function
CA	Certificate Authority
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GBA_ME	ME-based GBA
GBA_U	GBA with UICC-based enhancements
GUSS	GBA User Security Settings
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
Ks_int	Derived key in GBA_U which remains on UICC
Ks_ext	Derived key in GBA_U
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure
USS	GBA User Security Setting

===== BEGIN NEXT CHANGE =====

4.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the AKA protocol, and agree on session keys that are afterwards applied between UE and a~~n operator-controlled~~ Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using a suitable key derivation procedure. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause 4.5.2.

The BSF shall be able to acquire the GBA user security settings (GUSS) from the HSS.

4.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and a~~n operator-controlled~~ NAF can run some application specific protocol where the authentication of messages will be based on those session keys generated during the mutual authentication between UE and BSF.

General assumptions for the functionality of a~~n operator-controlled~~ NAF are:

- there is no previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire ~~an~~zero or more ~~(~~application-specific~~) user security setting~~ USSs from the HSS via the BSF;

NOTE 1: The NAF may be required to request one or more application-specific USS from the BSF. If the NAF fails to request a required USS this will result to an error.

- NAF shall be able to check lifetime of the shared key material.

===== BEGIN NEXT CHANGE =====

4.2.3 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS. There shall be at most one USS per application stored in the HSS. In the case where the subscriber has multiple subscriptions, i.e. multiple ISIM or USIM applications on the UICC, the HSS shall contain one or more subscriber profiles that can be mapped to one or more private identities, i.e. IMPIs and IMSIs.

~~Editor's note: Needed new subscriber profile parameters, i.e. GBA user security settings, are FFS.~~

The requirements on the HSS are:

- HSS shall provide the only persistent storage for G~~BA~~USSs;
- G~~BA~~USS shall be defined in such a way that interworking of different operators for standardised application profiles is possible;
- G~~BA~~USS shall be defined in such a way that profiles for operator specific applications and extensions to existing application profiles are supported without need for standardisation of these elements.
- GUSS shall be able to contain application-specific USSs that contain parameters that are related to identification or authorization information of one or more applications hosted by one ore more NAFs. Any other types of parameters are not allowed in the application-specific USS.

NOTE 1: The necessary subscriber profile data may be fetched by the NAF directly from HSS or from its local database using identity information provided by the application-specific USS.

NOTE 2: The HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber.

- GUSS shall be able to contain parameters intended for the BSF usage:
    - the type of the UICC the subscriber is issued (i.e., is it GBA_U aware or not, cf. subclause 5);
    - subscriber specific key lifetime.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or subsciber does not have GUSS then the BSF will use the default values in the BSF local policy defined by the particular MNO.

===== BEGIN NEXT CHANGE =====

4.4.3 Roaming

The roaming subscriber shall be able to utilize the bootstrapping function in the home network. The subscriber shall be able to utilize network application function that is in a visited network.

NOTE 1: A NAF in a visited network may be required to request one ore more application-specific USS from the home BSF (cf. subclause 4.4.6). If the NAF fails to do so, this will result to an error.

===== BEGIN NEXT CHANGE =====

4.4.6 Requirements on reference point Zn

The requirements for reference point Zn are:

- mutual authentication, confidentiality and integrity shall be provided;
- If the BSF and the NAF are located within the same operator's network, the Zn reference point shall be secured according to NDS/IP [13];
- If the BSF and the NAF are located in different operators' networks, the Zn' reference point between the D-Proxy and the BSF shall be secured using TLS as specified in RFC 2246 [6];

Editor’s Note: The TLS Certificate profiling needs to be completed and will be added into an Annex.

- The BSF shall verify that the requesting NAF is authorised;
- The NAF shall be able to send a key material request to the BSF, containing NAF's public hostname used by the UE's corresponding request. The BSF shall be able to verify that a NAF is authorized to use this hostname, i.e. the FQDN used by UE when it contacts the NAF;
- The BSF shall be able to send the requested key material to the NAF;
- The NAF shall be able to get a selected set of application-specific ~~user security settings~~USSs from the BSF, depending on the policy of the BSF and the application indicated in the request from the NAF over Zn;
- The NAF shall be able to indicate to the BSF the single application or several applications it requires ~~user security settings~~USS for;

NOTE 1: If some application needs only a subset of an application-specific ~~user security setting~~USS, e.g. only one IMPU, the NAF selects this subset from the complete set of ~~user security settings~~USSs sent from BSF.

- The BSF shall be able to be configured on a per NAF or per application basis if private subscriber identity and which ~~user security settings~~application-specific USSs may be sent to a NAF;
- The BSF shall be able to be configured in such a way that the BSF is able to decide on a per NAF basis if the NAF is required to request one or more application-specific USSs and if the requested USSs shall be present in GUSS, and to reject the request in case the conditions are not fulfilled.
- The BSF shall be able to indicate to the NAF the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 2: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

Editor's note: It is ffs which actions are to be taken over Zn when the BSF receives a user security settings update from the HSS over Zh.

===== BEGIN NEXT CHANGE =====

4.5.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause 4.5.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with an NAF the following steps are executed as depicted in figure 4.4.

UE starts communication over reference point Ua with the NAF:

- in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available),, the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:
    - if a key Ks for the selected UICC application is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause 4.5.2;
    - if no key Ks for the selected UICC application is available in the UE, the UE first agrees on a new key Ks with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF;

NOTE 1: If it is not desired by the UE to use the same Ks for the selected UICC application to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF;

- if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g. because the key's lifetime has expired, it shall send a suitable bootstrapping renegotiation request to the UE and terminates the protocol used over reference point Ua, see figure 4.5. The form of this indication depends on the particular protocol used over reference point Ua. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point Ub, as specified in clause 4.5.2, in order to obtain a new key Ks.

NOTE 2: To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see NOTE 2 of clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE 2 of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 3: If the shared key between UE and NAF is invalid , the NAF can set deletion conditions to the corresponding security association for subsequent removal.

- the UE supplies the B-TID to the NAF, in the form as specified in clause 4.3.2, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 4: The UE may adapt the key material Ks_NAF to the specific needs of the reference point Ua. This adaptation is outside the scope of this specification.

- when the UE is powered down, or when the UICC is removed, any keys Ks and Ks_NAF shall be deleted from storage;
- when a new Ks is agreed over the reference point Ub and a key Ks_NAF, derived from one NAF_Id, is updated, the other keys Ks_NAF, derived from different values NAF_Id, stored on the UE shall not be affected;

NAF starts communication over reference point Zn with BSF

- The NAF requests key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua. If the NAF has several FQDNs, which may be used in conjunction with this specification, then the NAF shall transfer in the request over Zn the same FQDN, which was used over Ua (see NOTE 2 on key derivation in this clause);

- The NAF may also request one or more application-specific ~~user security settings~~USSs for the applications, which the request received over Ua from UE may access;
- With the key material request, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able verify that NAF is authorized to use that hostname;
- The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key Ks_NAF, as well as the lifetime of that key, and the requested application-specific USSs if available. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.

NOTE 5: The NAF shall adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.

NOTE 6: The BSF may require that the NAF requests one or more application-specific USSs (cf. subclause 4.4.6). If one or more of these required settings are not requested by the NAF, the BSF shall indicate this in the reply to the NAF.

- The BSF may also send the private user identity (IMPI) and requested ~~user security settings~~USSs to NAF according to the BSF's policy;

NAF continues with the protocol used over the reference point Ua with the UE.

Once the run of the protocol used over reference point Ua is completed the purpose of bootstrapping is fulfilled as it enabled UE and NAF to use reference point Ua in a secure way.

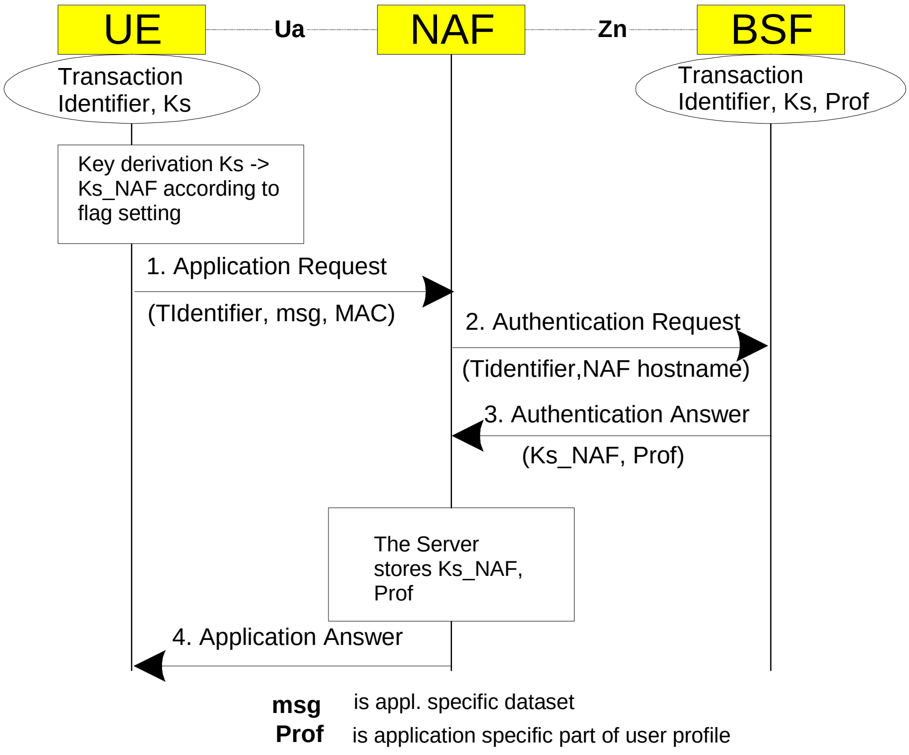
UE
Ua
NAF
Zn
BSF
Transaction Identifier, Ks
Transaction Identifier, Ks, Prof
Key derivation Ks -> Ks_NAF according to flag setting
1. Application Request
(TIdentifier, msg, MAC)
2. Authentication Request
(Tidentifier,NAF hostname)
3. Authentication Answer
(Ks_NAF, Prof)
The Server stores Ks_NAF, Prof
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

Figure 4.4: The bootstrapping usage procedure

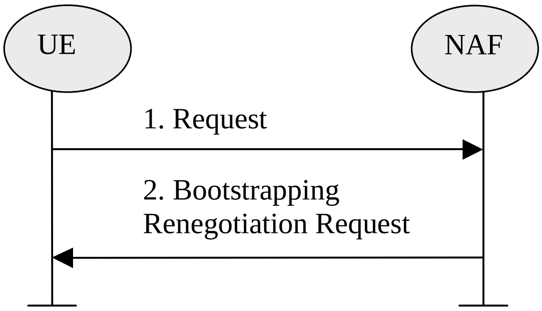

Figure 4.5: Bootstrapping renegotiation request

===== BEGIN NEXT CHANGE =====

5.3.3 Procedures using bootstrapped Security Association

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use shared keys obtained by means of the GBA. If the UE does not know whether to use GBA with this NAF, it uses the Initiation of Bootstrapping procedure described in clause 5.3.1.

Once the UE and the NAF have established that they want to use GBA then every time the UE wants to interact with a NAF the following steps are executed as depicted in figure 5.3.

Next, the UE and the NAF have to agree, which type of keys to use, Ks_ext_NAF or Ks_int_NAF, or both. The default is the use of Ks_ext_NAF only. This use is also supported by MEs and NAFs, which are GBA_U unaware. If Ks_int_NAF, or both Ks_ext and Ks_int are to be used, this use has to be agreed between UE and NAF prior to the execution of the procedure described in the remainder of this clause 5.3.3. Any such agreement overrules the default use of the keys. How this agreement is reached is application-specific and is not within the scope of this document.

NOTE 1: This agreement may be mandated by the specification, which defines the Ua reference point between UE and NAF, e.g. TS 33.246 for the use of GBA in MBMS, or negotiated by the NAF and the UE over the Ua reference point, or reached by configuration.

Editors' Note: The support of unaware GBA_U MEs, which are GBA_ME aware only is FFS.

In general, UE and NAF will not yet share the key(s) required to protect the Ua reference point. If they do not, the UE proceeds as follows:

- if Ks_ext_NAF is required and a key Ks_ext for the selected UICC application is available in the UE, the UE derives the key Ks_ext_NAF from Ks_ext, as specified in clause 5.3.2;
- if Ks_int_NAF is required and a key Ks_int for the selected UICC application is available in the UICC, the ME requests the UICC to derive the key Ks_int_NAF from Ks_int, as specified in clause 5.3.2;

NOTE 2: If it is not desired by the UE to use the same Ks_ext/int for the selected UICC application to derive more than one Ks_ext/int_NAF then the UE should first agree on new keys Ks_ext and Ks_int with the BSF over the Ub reference point, as specified in clause 5.3.2, and then proceeds to derive Ks_ext_NAF or Ks_int_NAF, or both, as required.

- if Ks_ext and Ks_int for the selected UICC application are not available in the UE, the UE first agrees on new keys Ks_ext and Ks_int with the BSF over the Ub reference point, as specified in clause 5.3.2, and then proceeds to derive Ks_ext_NAF or Ks_int_NAF, or both, as required;
- if the NAF shares a key with the UE, but the NAF requires an update of that key, it shall send a suitable bootstrapping renegotiation request to the UE and terminate the protocol used over Ua reference point. The form of this indication depends on the particular protocol used over Ua reference point. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over Ub, as specified in clause 5.3.2, in order to obtain new keys.

NOTE 3: If the shared keys between UE and NAF become invalid, the NAF can set deletion conditions to the corresponding security association for subsequent removal.

NOTE 4: If it is not desired by the NAF to use the same Ks to derive more than one Ks_int/ext_NAF then the NAF should always reply to the first request sent by a UE by sending a key update request to the UE.

UE and NAF can now start the communication over Ua reference point using the keys Ks_ext_NAF or Ks_int_NAF, or both, as required. They proceed as follows:

- The UE supplies the B-TID to the NAF, as specified in clause 5.3.2, to allow the NAF to retrieve the corresponding keys from the BSF

NOTE 5: To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (cf. NOTE 2 of clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE 2 of clause 4.5.2 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

NOTE 6: The UE may adapt the keys Ks_ext_NAF or Ks_int_NAF to the specific needs of the Ua reference point. This adaptation is outside the scope of this specification.

- when the UE is powered down, or when the UICC is removed, any GBA_U keys shall be deleted from storage in the ME. There is no need to delete keys Ks_int and Ks_int_NAF from storage in the UICC;

NOTE 7: After each run of the protocol over the Ub reference point, new keys Ks_ext and Ks_int, associated with a new B-TID, are derived in the UE according to clause 5.3.2, so that it can never happen, that keys Ks_ext and Ks_int with different B-TIDs simultaneously exist in the UE.

- When new keys Ks_ext and Ks_int are agreed over the Ub reference point and new NAF-specific keys need to be derived for one NAF_Id, then both, Ks_ext_NAF and Ks_int_NAF (if present), shall be updated for this NAF_Id, but further keys Ks_ext_NAF or Ks_int_NAF relating to other NAF_Ids, which may be stored on the UE, shall not be affected;

NOTE 8: This rule ensures that the keys Ks_ext_NAF and Ks_int_NAF are always in synch at the UE and the NAF.

NAF now starts communication over the Zn reference point with the BSF.

- The NAF requests from the BSF the keys corresponding to the B-TID, which was supplied by the UE to the NAF over the Ua reference point. If the NAF is GBA_U aware it indicates this by including a corresponding flag in the request. If the NAF has several FQDNs, which may be used in conjunction with this specification, then the NAF shall transfer in the request over Zn the same FQDN, which was used over Ua (see note above on key derivation in this clause).

- The NAF may also request one or more application-specific ~~user security settings~~USSs for the applications, which the request received over Ua from UE may access;

- With the keys request over the Zn reference point, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able to verify that NAF is authorized to use that hostname.

- The BSF derives the keys Ks_ext_NAF, and Ks_int_NAF (if additionally required), as specified in clause 5.3.2. If the NAF indicated in its request that it is GBA_U aware, the BSF supplies to NAF both keys, Ks_ext_NAF, and Ks_int_NAF, otherwise the BSF supplies only Ks_ext_NAF. In addition, the BSF supplies the lifetime time of these keys, and the requested application-specific USSs if available. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request (See figure 4.5) to the UE;

NOTE 9: The NAF may adapt the keys Ks_ext_NAF and Ks_int_NAF to the specific needs of the Ua reference point in the same way as the UE did. This adaptation is outside the scope of this specification.

NOTE 10: The BSF may require that the NAF requests one or more application-specific USSs (cf. subclause 4.4.6). If one or more of these required settings are not requested by the NAF, the BSF shall indicate this in the reply to the NAF.

- The BSF may also send the private user identity (IMPI) and requested ~~user security settings~~USSs to NAF according to the BSF's policy.

The NAF now continues with the protocol used over the Ua reference point with the UE.

Once the run of the protocol used over Ua reference point is completed the purpose of bootstrapping is fulfilled as it enabled the UE and NAF to use Ua reference point in a secure way.

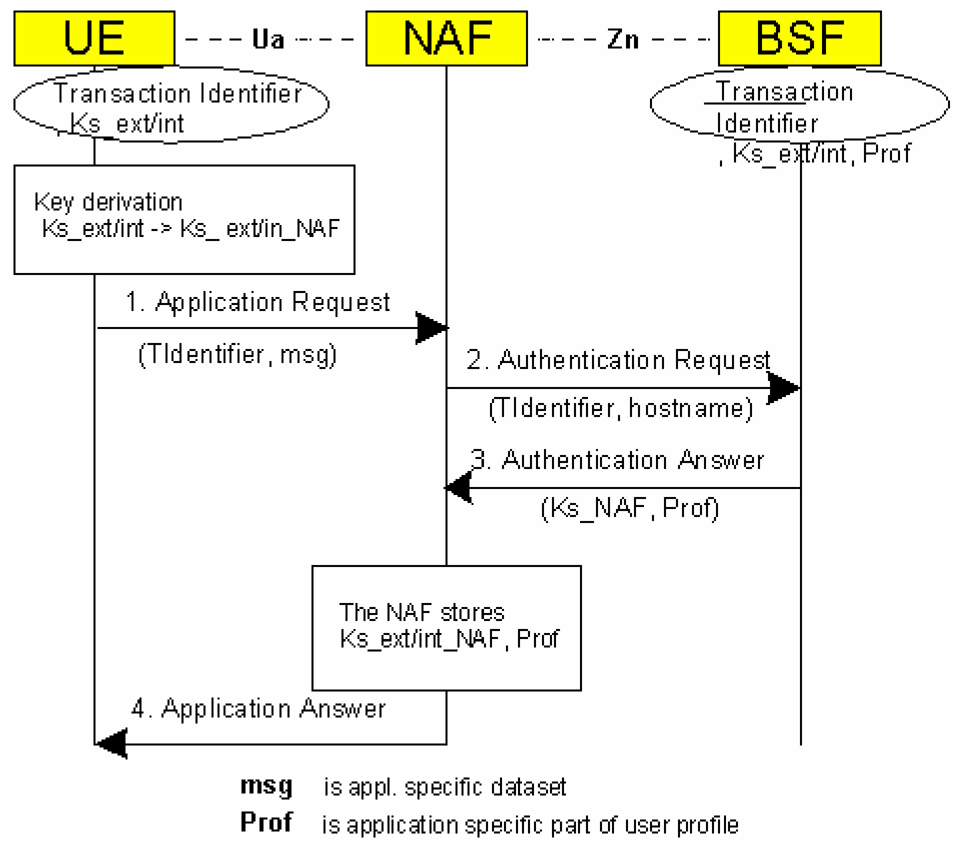
UE
Ua
NAF
Zn
BSF
Transaction Identifier, Ks_ext/int
Transaction Identifier, Ks_ext/int, Prof
Key derivation Ks_ext/int -> Ks_ext/in_NAF
1. Application Request
(TIdentifier, msg)
2. Authentication Request
(TIdentifier, hostname)
3. Authentication Answer
(Ks_NAF, Prof)
The NAF stores Ks_ext/int_NAF, Prof
4. Application Answer
msg is appl. specific dataset
Prof is application specific part of user profile

Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

===== END CHANGE =====