3GPP TSG SA WG3 Security — S3#32 **S3-040041**
09 - 13 February 2004
Edinburgh, Scotland, UK

Source:	**Siemens**
Title:	**Key handling in the UE in a Generic Bootstrapping Architecture - Pseudo-CR**
Agenda Item:	**6.9.2 (GBA)**
Document for:	**Discussion and decision**

Abstract

In the current version of the Generic Bootstrapping Architecture specification (TS 33.220 v100), key handling issues are not explicitly addressed. This contribution proposes text regarding the handling for the keys Ks_NAF shared between UE and NAFs. In particular, it is proposed that the keys are deleted when the UE is powered down. A pseudo-CR to TS 33.220 v100 is also provided in this contribution.

1. Reason for proposed change to TS 33.220 v100

Key handling issues are not explicitly addressed in TS 33.220 v100. For the UE, rules regarding the handling of keys are needed in order to avoid security problems.

A UE and a NAF share a key Ks_NAF. As the UE may communicate with various NAFs over time, the UE will have several keys Ks_NAF in storage at a given time. (Otherwise, the UE would have to get a new key each time it wants to get back to a NAF with which it communicated before.) These shall be handled in the UE according to the following rules:

- *When the UE is powered down, or when the UICC is removed, any keys Ks and Ks_NAF shall be deleted from storage*. This is to minimise the risk that keys can be accessed in semi-permanent storage on the terminal. It is also important for terminals controlled by changing users (with different UICCs), e.g. rental terminals.
- *When a new Ks is obtained over the Ub interface and a key Ks_NAF, derived from one NAF_Id_n, is updated, the other keys Ks_NAF, derived from different values NAF_Id_n, stored on the UE shall not be affected.* An update of a Ks_NAF may be requested by a NAF, according to TS 33.220 v100, section 4.3.3. The proposed rule is to avoid an avalanche of key updates caused by this request.
- *KDF shall be implemented in the ME*. This seems quite sufficient from a security point of view, as the Ks is a session key which will be deleted when the UE is powered down..

The next section contains a pseudo-CR to TS 33.220 v100, implementing the changes proposed in this section.

2. Pseudo-CR

4.3.2 Bootstrapping procedures

When a UE wants to interact with an NAF, and it knows that bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 4)

Editor's note: Zh interface related procedure will be added here in future development. It may re-use Cx interface that is specified in TS 29.228.

Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a key update indication from the NAF, or when the lifetime of the key has expired (cf. subclause 4.3.3).

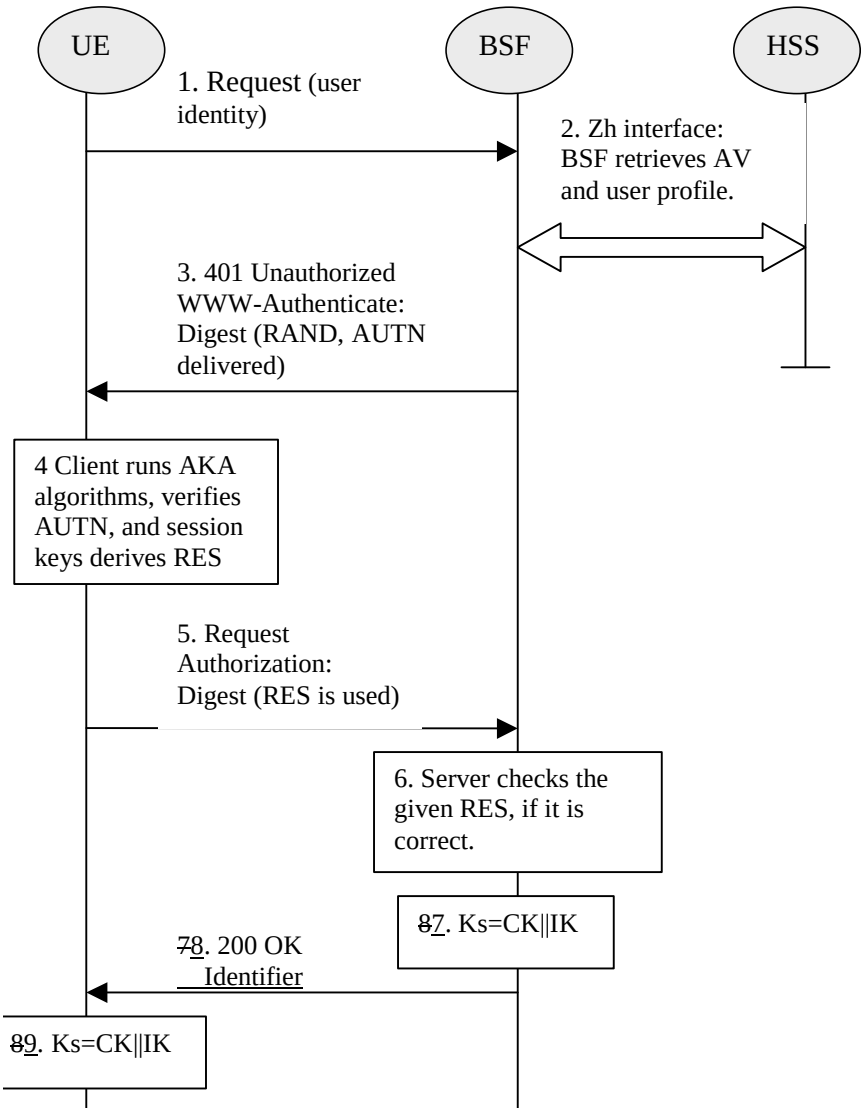

Figure 4: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the user profile and a challenge, i.e. the Authentication Vector (AV, AV = RAND||AUTN||XRES||CK||IK) over Zh interface from the HSS.
3. Then BSF forwards the RAND and AUTN to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.
4. The UE calculates the message authentication code (MAC) so as to verify the challenge from authenticated network; the UE also calculates CK, IK and RES. This will result in session keys IK and CK in both BSF and UE.
5. The UE sends request again, with the Digest AKA RES as the response to the BSF.
6. If the RES equals to the XRES that is in the AV, the UE is authenticated.
7. BSF generates key material Ks by concatenating CK and IK. Ks is used to derive the key material Ks_NAF. Ks_NAF is used for securing the Ua interface.

8. The BSF shall send 200 OK message and shall supply a transaction identifier to the UE to indicate the success of the authentication. The BSF may also supply the parameter *n* used to determine the NAF_Id_n (cf. previous bullet) to the UE over the Ub interface. If the parameter *n* is not supplied then no key derivation is performed, i.e. Ks = Ks_NAF.

9. The key material Ks is generated in UE by concatenating CK and IK. The Ks is used to derive the key material Ks_NAF. Ks_NAF is used for securing the Ua interface.

Ks_NAF is computed as Ks_NAF = KDF (Ks, key derivation parameters), where KDF is a suitable key derivation function, and the key derivation parameters include the user's IMSI, the NAF_Id_n and RAND. KDF shall be implemented in the ME.

The NAF_Id_n consists of the n rightmost domain labels in the DNS name of the NAF, separated by dots (n= 1, ..., 7). For n = 0, NAF_Id_n equals the full DNS name of the NAF. The next bullet specifies how the UE obtains n.

NOTE: This note gives an example how to obtain the NAF_Id_n: if the DNS name of the NAF is "server1.presence.bootstrap.operator.com", and n = 3, then NAF_Id_n = " bootstrap.operator.com".

Editor's note: The definition of the KDF and the possible inclusion of further key derivation parameters is left to ETSI SAGE.

4.3.3 Procedures using bootstrapped Security Association

After UE is authenticated with the BSF, every time the UE wants to interact with an NAF the following steps are executed as depicted in figure 5.

UE starts communication over Ua interface with the NAF

- In general, UE and NAF will not yet share the key(s) required to protect Ua interface. If they already do, there is no need for NAF to retrieve the key(s) over Zn interface.

- If the NAF shares a key with the UE, but an update of that key it sends a suitable key update request to the UE and terminates the protocol used over Ua interface. The form of this indication may depend on the particular protocol used over Ua interface and is ffs.

- It is assumed that UE supplies sufficient information to NAF, e.g. a transaction identifier, to allow the NAF to retrieve specific key material from BSF.

- The UE derives the keys required to protect the protocol used over Ua interface from the key material, as specified in clause 4.3.2.

NOTE 1: The UE may adapt the key material Ks_NAF to the specific needs of the Ua interface. 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 obtained over the Ub interface and a key Ks_NAF, derived from one NAF_Id_n, is updated, the other keys Ks_NAF, derived from different values NAF_Id_n, stored on the UE shall not be affected.

NAF starts communication over Zn interface with BSF

- The NAF requests key material corresponding to the information supplied by the UE to the NAF (e.g. a transaction identifier) in the start of the protocol used over Ua interface.

- The BSF derives the keys required to protect the protocol used over Ua interface from the key material and the key derivation parameters, as specified in clause 4.3.2, and supplies to NAF the requested key material. If the key identified by the transaction identifier 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 key update request to the UE.

NOTE 2: The NAF may adapt the key material Ks_NAF to the specific needs of the Ua interface in the same way as the UE did. This adaptation is outside the scope of this specification.

NAF continues with the protocol used over Ua interface with UE.

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

Editor's note: Message sequence diagram presentation and its details will be finalized later.

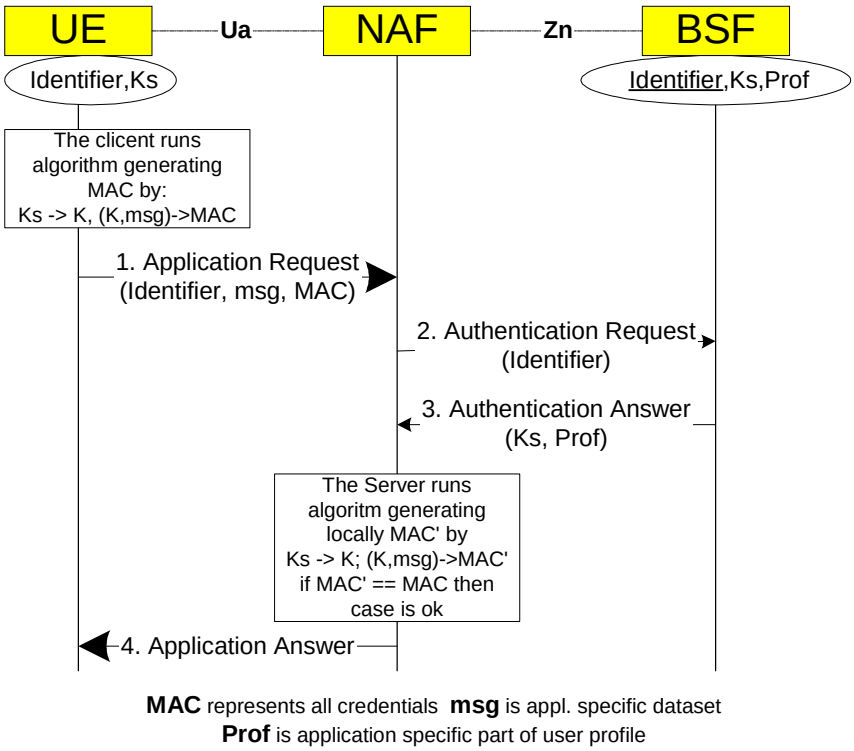

Figure 5: The bootstrapping usage procedure