3GPP TSG SA WG3 Security — S3#31 **S3-030729**
18 - 21 November 2003
Munich, Germany

Source: **Nokia**

Title: **UE triggered unsolicited push from BSF to NAFs**

Document for: **Discussion/Decision**

Agenda Item: **6.9 (GAA)**

1. INTRODUCTION

This contribution discusses a possibility for UE to trigger BSF to do an unsolicited push of transaction identifier (TID), NAF specific shared secret (Ks_naf), and optional subscriber profile information (TID, Ks_naf, and the profile are later referred as "bootstrapping information") to one or more NAFs. This would simplify procedures during shared secret usage over Ua interface (between UE and NAF) since if NAF already has received the bootstrapping information and it does not need to use Zn interface fetch this information.

2. DISCUSSION

2.1 Use cases for the proposed procedures

2.1.1 Use case 1: Triggering during bootstrapping procedure

In order for UE to trigger BSF to do an unsolicited push of bootstrapping information to one or more NAFs, a list of NAF_IDs is inserted to the initial bootstrapping request. In the list there can be zero or more NAF_IDs present. NAF_IDs are known by the BSF so that it knows to which NAFs the TID, Ks_naf, and NAF type specific profile information is to be pushed to the NAFs.

2.1.2 Use case 2: Triggering of previous bootstrapping information

UE may also trigger BSF to do unsolicited push of bootstrapping information without doing full bootstrapping procedure over Ub interface. In this case, the TID from previous bootstrapping procedure is inserted to the initial bootstrapping request. In this case, there must be one or more NAF_IDs present. If BSF finds the TID acceptable (i.e., the previous bootstrapping is not too old or it is otherwise valid), it will push the previous bootstrapping information to the NAFs identified by NAF_IDs. If BSF finds the TID invalid, full bootstrapping procedure is executed.

2.2 NAF_ID considerations

Requirements for NAF_ID:

- NAF_ID shall be in a format, which is easily discovered or known by the UE.
- NAF_ID shall be globally unique to identify a NAF.

Solution for NAF_ID format:

- **FQDN**[1] of the NAF. It is easily discovered or known by the UE and uniquely identifies the NAF.

[1] Fully qualified domain name (FQDN) consists of a host and domain name, including top-level domain.

2.3 Procedure details

Figure 1 describes the bootstrapping procedure where UE triggers the pushing of bootstrapping information from BSF to one or more NAFs.

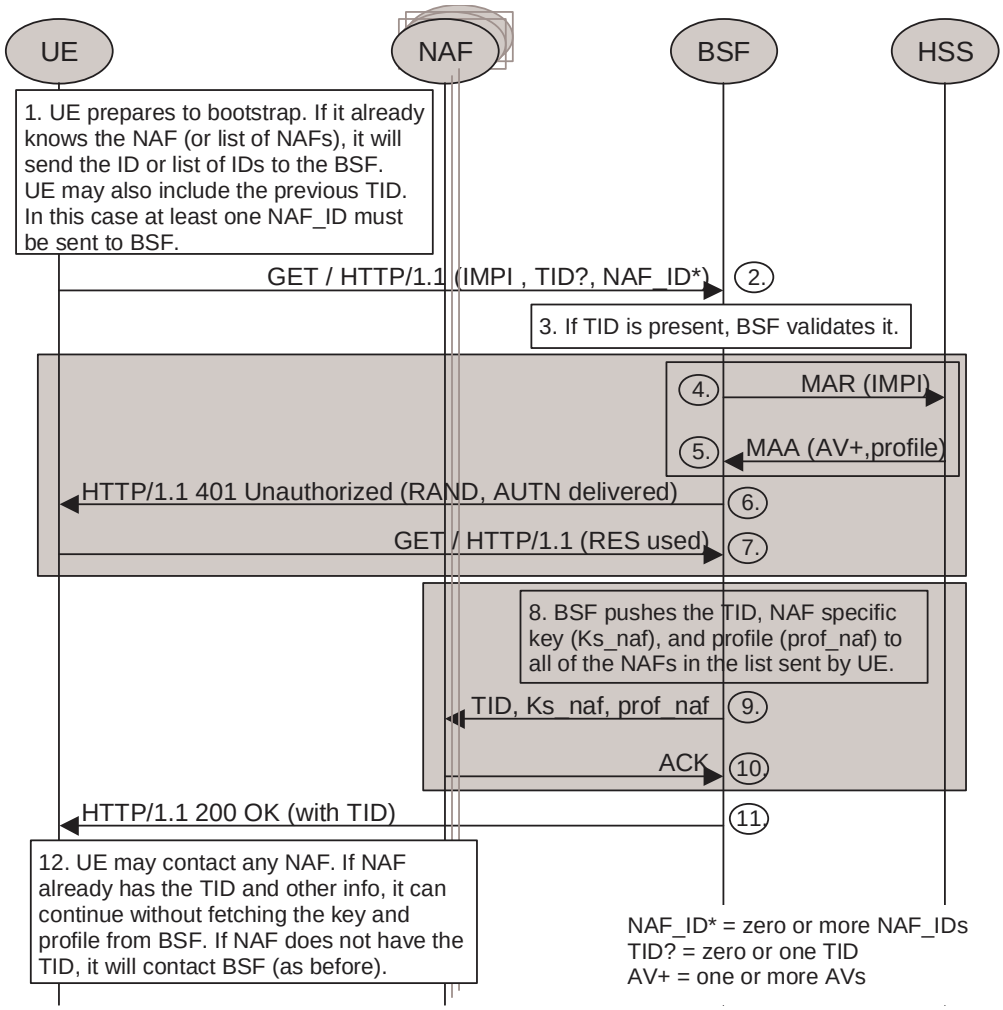


Figure 1. UE triggered push of bootstrapping information from BSF to NAF.

1. UE prepares to do bootstrapping procedure or trigger a push operation from BSF to NAF using a previous bootstrapping info. If UE already knows the NAF and multiple NAFs, it prepares a list of NAF_IDs to be inserted in to the initial bootstrapping message. If UE is triggering BSF to push previous bootstrapping info to a NAF or multiple NAFs, at least one NAF_ID must be sent to BSF.

2. UE send initial HTTP request to BSF with IMPI, optional TID, and zero or more NAF_IDs.

3. If TID was present in the message, BSF checks whether the corresponding bootstrapping info is still valid. If it is valid, procedure continues from step 8.

Note: Steps 4-7 happen only if BSF decides that the TID is not valid or no TID was present in the initial bootstrapping request in step 2.

4-5. (Optional) BSF fetches authentication vectors and profile information from HSS. Optionally, if BSF already has authentication vectors for the UE it may either just request for profile update, or skip steps 4 and 5 all together.

6-7. (Optional) Ordinary HTTP Digest AKA steps are done and new bootstrapping info is established.

Note: Steps 8-10 are optional and happen only if one or more NAF_IDs were listed in the initial bootstrapping request in step 2.

8. (Optional) If NAF_IDs were present in the initial HTTP request (step 2), then BSF pushed the bootstrapping info to the listed NAFs.

9. (Optional) NAF specific bootstrapping info is pushed to a NAF by BSF.

10. (Optional) NAF acknowledges that the bootstrapping info was received and stored.

11. HTTP response 200 OK is sent to the UE to indicate that the procedure was successful: either valid TID was sent in step 2 or the HTTP Digest AKA was successful. There is no indication whether if the push operation was successful or not. The response contains also the new TID if full bootstrapping procedure was done.

12. UE contacts any NAF it desires. If the corresponding NAF already contains the bootstrapping info belonging to the UE, it does not need to fetch the bootstrapping info over Zn interface.

2.4 Analysis

If the unsolicited push of bootstrapping information to selected NAFs is done, it simplifies the procedures NAF needs to do during shared key usage over Ua interface; when UE wishes to use Ua interface with a NAF, it would already have the related TID, shared secret, and profile information in its cache and there would be no need to get the bootstrapping information from the BSF online.

The optional values (i.e., TID and NAF_IDs) are not integrity protected in the initial bootstrapping request (step 2 in Figure 1). Thus, an active attacker may have opportunities to change the corresponding values in use case 2. In use case 1, the attack can be avoided by sending the NAF identifiers in the second HTTP GET message (step 7 in Figure 1) where the parameters would be protected using HTTP Digest AKA integrity protection (i.e., qop="auth-int"). However, if an attacker changes these values, it would merely just result to a denial of service attack since BSF would send the bootstrapping info only to valid and well-known NAFs, and ignore bad or unknown NAF_IDs. Afterwards, when UE accesses a NAF, which has not received the bootstrapping info, NAF will fetch this information from BSF over Zn interface as specified in [TS GBA].

3. PROPOSAL

We propose to add the unsolicited push mechanism described in this contribution to the bootstrapping procedure described in 3GPP TS 33.220 [TS GBA]. Use case 2 may be specified as optional.

REFERENCES

[TS GBA] Draft 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture", Rel-6.

******* BEGIN CHANGE *******

4 Generic Bootstrapping Architecture

The 3GPP authentication infrastructure, including the 3GPP Authentication Centre (AuC), the USIM, and the 3GPP AKA protocol run between them, is a very valuable asset of 3GPP operators. It has been recognised that this infrastructure could be leveraged to enable application functions in the network and on the user side to communicate in situations where they would not be able to do so without the support of the 3GPP authentication infrastructure. Therefore, 3GPP can provide the "bootstrapping of application security" to authenticate the subscriber by defining a generic bootstrapping function based on AKA protocol.

4.1 Requirements and principles for bootstrapping

Editor's note: The description of AKA bootstrapping shall be added here.

- The bootstrapping function shall not depend on the particular network application function
- The server implementing the bootstrapping function needs to be trusted by the home operator to handle authentication vectors.
- The server implementing the network application function needs only to be trusted by the home operator to handle derived key material.
- It shall be possible to support network application functions in the operator's home network
- The architecture shall not preclude the support of network application function in the visited network, or possibly even in a third network.
- To the extent possible, existing protocols and infrastructure should be reused.
- In order to ensure wide applicability, all involved protocols are preferred to run over IP.

4.1.1 Access Independence

Bootstrapping procedure is access independent. Bootstrapping procedure requires IP connectivity from UE.

4.1.2 Authentication methods

Authentication method that is used to authenticate the bootstrapping function must be dependent on cellular subscription. In other words, authentication to bootstrapping function shall not be possible without valid cellular subscription. Authentication shall be based on AKA protocol.

4.1.3 Roaming

The roaming subscriber shall be able to utilize the bootstrapping function in home network.

Editor's note: For the first phase of standardisation, only the case is considered where bootstrapping server functionality and network application function are located in the same network as the HSS. In later phases, other configurations may be considered.

4.1.4 Requirements on Ub interface

The requirements for Ub interface are:

- The BSF shall be able to identify the UE.
- The BSF and the UE shall be able to authenticate each other based on AKA.
- The BSF shall be able to send a transaction identifer to UE.

- The UE shall be able to indicate to the BSF, that it wants to push key material and subscriber profile during bootstrapping procedure to one or more NAFs.

- The UE shall be able to indicate to the BSF, that it wants to push key material and subscriber profile from previous bootstrapping procedure to one or more NAFs.

Editor's note: The format of NAF identifier is ffs. NAF identifier must uniquely identity a NAF and it must be easily discovered by the UE. One solution for NAF identifier is fully qualified domain name (FQDN).

4.1.5 Requirements on Zh interface

The requirements for Zh interface are:

- The BSF shall be able to communicate securely with the subscriber's HSS.

Editor's note: this requirement is fulfilled automatically if BSF and HSS are in same operator's network.

- The BSF shall be able to send bootstrapping information request concerning a subscriber.
- The HSS shall be able to send authentication vectors to the BSF in batches.
- The HSS shall be able to send the subscriber's GAA profiles to the BSF.

Editor's note: it's ffs how to proceed in the case where profile is updated in HSS after profile is forwarded. The question is whether this profile change should be propagated to BSF.

- No state information concerning bootstrapping shall be required in the HSS.
- All procedures over Zh interface shall be initiated by the BSF.
- It is preferred to reuse existing specifications if possible.
- The number of different interfaces to HSS should be minimized.

4.1.6 Requirements on Zn interface

The requirements for Zn interface are:

- NAF shall be able to communicate securely with a subscriber's BSF.
- The NAF shall be able to send a key material request to the BSF.
- The BSF shall be able to send the requested key material to the NAF.
- The NAF shall be able to get the subscriber profile from BSF.

- The BSF shall be able to push key material and subscriber profile to the NAF.

Editor's note: in later phases there is an additional requirement that the NAF and the BSF may be in different operators' networks.

4.2 Bootstrapping architecture

4.2.1 Reference model

Figure 1 shows a simple network model of the entities involved in the bootstrapping approach, and the protocols used among them.

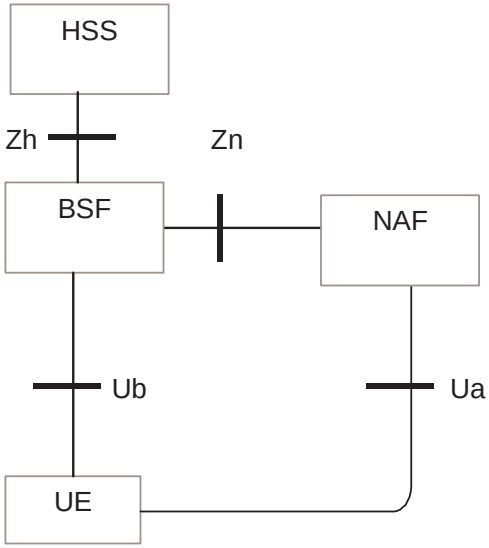


Figure 1: Simple network model for bootstrapping

Figure 2 illustrates a protocol stacks structure in network elements that are involved in bootstrapping of application security from 3G AKA and support for subscriber certificates.

Editor's note: The current protocol stack figure is placed here as a holder. The actual protocols will be defined later.

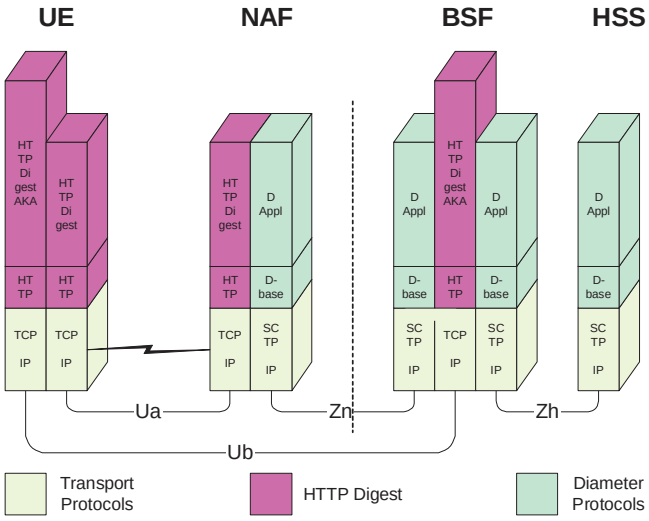


Figure 2: Protocol stack architecture

4.2.2 Network elements

4.2.2.1 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 an operator-controlled network application function (NAF). The key material must be generated specifically for each NAF independently.

Editor's note: key generation for NAF is ffs. Potential solutions may include:

- Separate run of HTTP Digest AKA over Ub interface for each request of key material from a NAF
- Derivation of NAF-specific keys in BSF

4.2.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and an operator-controlled network application function (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 an operator-controlled network application function (NAF):

- There is no previous security association between the UE and the NAF.
- NAF shall able to locate and communicate securely with subscriber's BSF.
- NAF shall be able to acquire a shared key material established between UE and the bootstrapping server function (BSF) during running application-specific protocol.
- BSF shall be able to push a shared key material established between UE and the BSF to NAF before running application-specific protocol.

4.2.2.3 HSS

HSS shall store new parameters in subscriber profile related to the usage of bootstrapping function. Possibly also parameters related to the usage of some network application function are stored in HSS.

Editor's note: Needed new parameters are FFS.

4.2.2.4 UE

The required new functionalities from UE are:

- The support of HTTP Digest AKA protocol,
- The capability to derive new key material to be used with the protocol over Ua interface from CK and IK, and
- Support of NAF specific application protocol (see [5]).

4.2.3 Reference points

4.2.3.1 Ub interface

The reference point Ub is between the UE and the BSF. The functionality is radio access independent and can be run in both CS and PS domains.

Editor's notes: The solution for CS domain is ffs.

4.2.3.1.1 Functionality

Reference point Ub provides mutual authentication between the UE and the BSF entities. It allows the UE to bootstrap the session keys based on the 3G infrastructure. The session key as result of key agreement functionality, is used to support further applications e.g. certificate issuer.

4.2.3.1.2 Protocol

Ub interface is in format of HTTP Digest AKA, which is specified in [4]. It is based on the 3GPP AKA [2] protocol that requires information from USIM and/or ISIM. The interface to the USIM is as specified for 3G [1].

4.2.3.2 Ua interface

Ua interface is the application protocol which is secured using the keys material agreed between UE and BSF as a result of the run of HTTP Digest AKA over Ub interface. For instance, in the case of support for subscriber certificates [5], it is a protocol, which allows the user to request certificates from the NAF. In this case NAF would be the PKI portal.

4.2.3.3 Zh interface

Zh interface is used between the BSF and the HSS to allow the BSF to fetch the required authentication information and subscriber profile information from the HSS. The interface to the 3G Authentication Centre is HSS-internal, and it need not be standardised as part of this architecture.

4.2.3.4 Zn interface

Zn interface is used by the NAF to fetch the key material agreed during previous HTTP Digest AKA protocol run over Ub interface from the BSF. Zn interface is also used by the BSF to push the key material agreed during current or previous HTTP Digest AKA protocol run over Ub interface to one or more NAFs. It may also be used to fetch subscriber profile information from BSF.

4.3 Procedures

This chapter specifies in detail the format of the bootstrapping procedure that is further utilized by various applications. It contains the AKA authentication procedure with BSF, and latter the key material generation procedure.

4.3.1a Initiation of bootstrapping

When a UE wants to interact with an NAF, but it does not know if bootstrapping procedure is required, it shall contact NAF for further instructions (see Figure 3).

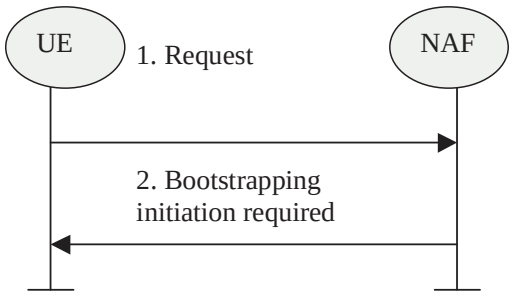


Figure 3: Initiation of bootsrapping

1. UE starts communication over Ua interface with the NAF without any bootstrapping related parameters.
2. If the NAF require bootstrapping but the request from UE does not include bootstrapping related parameters, NAF replies with a bootstrapping initiation message. The form of this indication may depend on the particular Ua interface and is ffs.

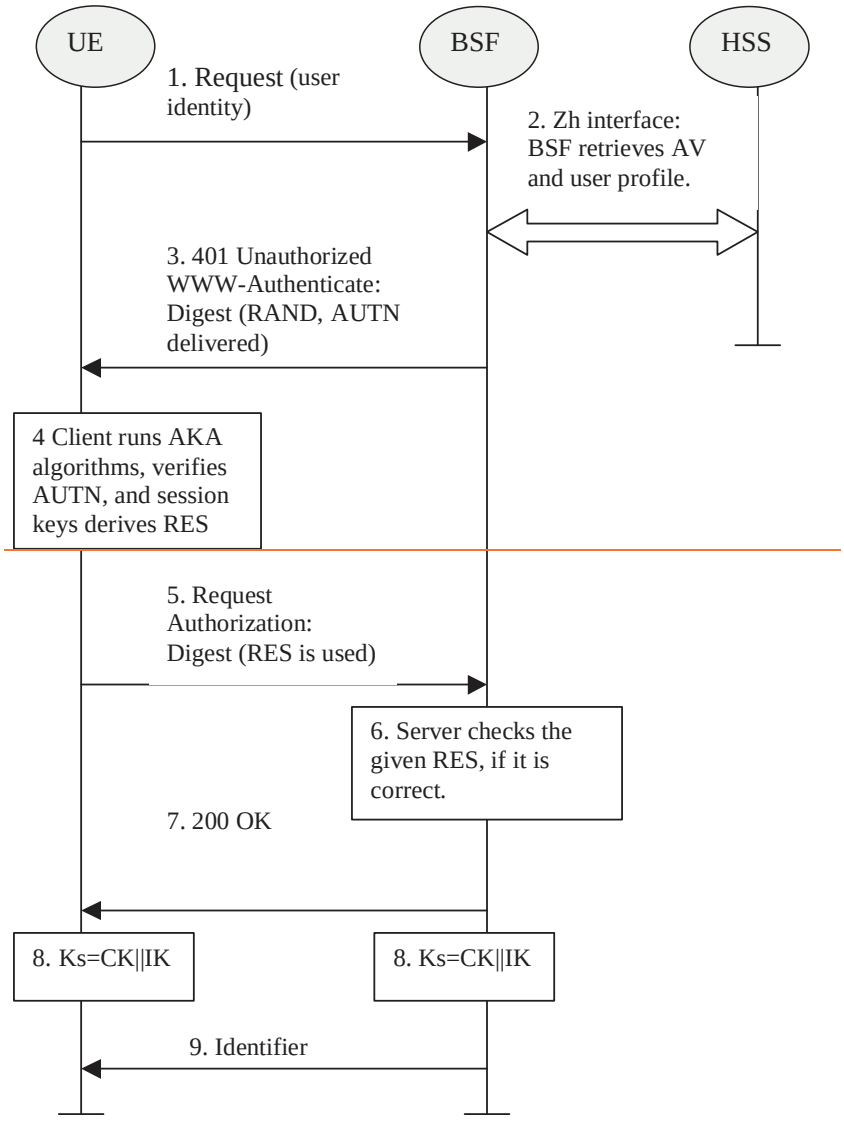
Editor's notes: If the protocol over Ua interface is based on HTTP, then NAF can initiate the bootstrapping procedure by using HTTP status codes (e.g. 401 Unauthorized).

4.3.1 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 notes: 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 (cf. subclause 4.3.2).


UE
BSF
HSS
1. Request (user identity)
2. Zh interface: BSF retrieves AV and user profile.
3. 401 Unauthorized WWW-Authenticate: Digest (RAND, AUTN delivered)
4 Client runs AKA algorithms, verifies AUTN, and session keys derives RES
5. Request Authorization: Digest (RES is used)
6. Server checks the given RES, if it is correct.
7. 200 OK
8. Ks=CK||IK
8. Ks=CK||IK
9. Identifier

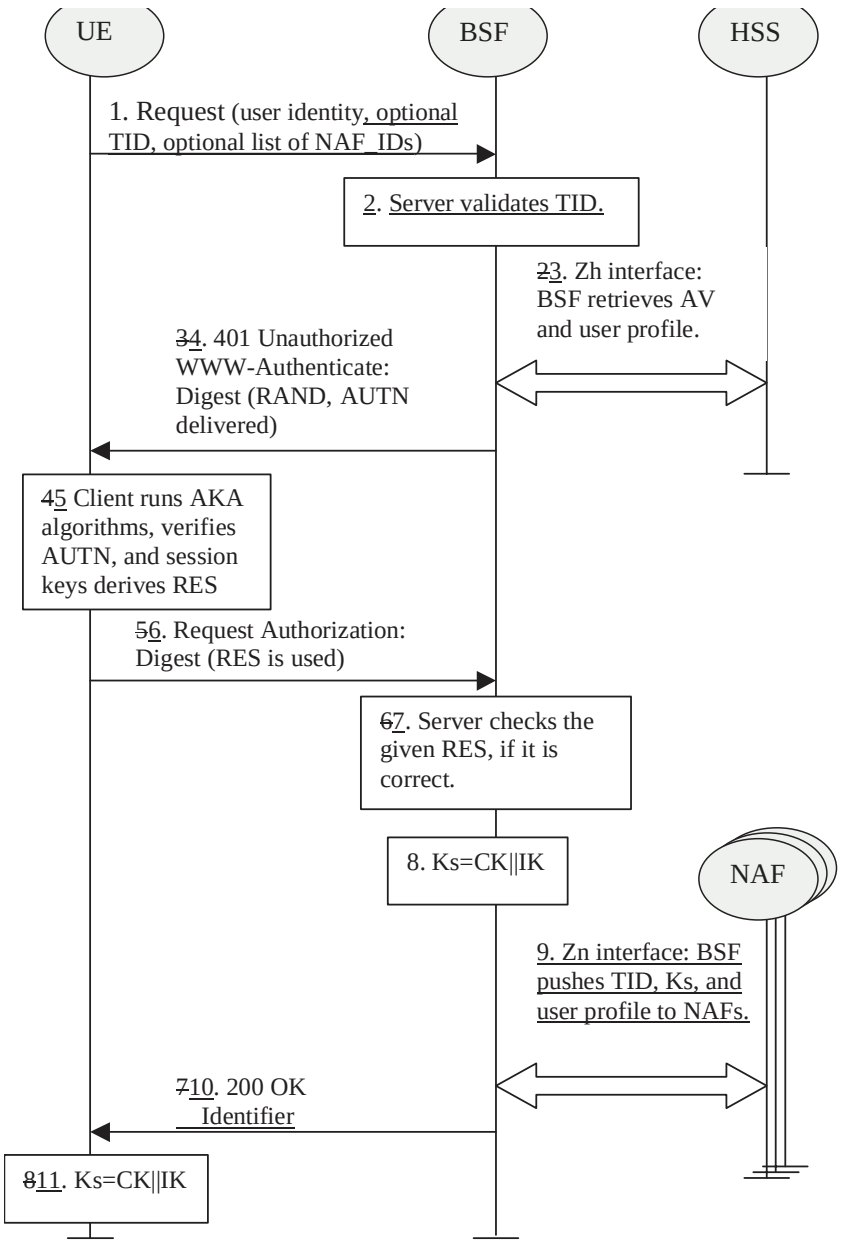


Figure 4: The bootsrapping procedure

1. The UE sends an HTTP request towards the BSF. The requst contains user identity. Request may contain a TID from UE's previous bootstrapping procedure, and list of NAF identifiers. If TID is present, then request shall contain at least one NAF identifier.

2. If TID is present in the request, BSF validates it. If TID is valid, BSF skips steps 3-8.

23. 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.

34. 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.

45. 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.

56. The UE sends request again, with the Digest AKA RES as the response to the BSF.

~~6~~7. If the RES equals to the XRES that is in the AV, the UE is authenticated.

8. BSF generates key material Ks by concatenating CK and IK. Ks is used for securing the Ua interface.

9. If NAF identifiers were present in the initial bootstrapping request, BSF pushes the TID, Ks, and NAF specific user profile to NAFs identified by the NAF identifiers.

Editor's note: The key provided to NAF may be generated from key material Ks making it a NAF specific key Ks_naf. How the key generation is done is ffs.

~~7~~10. The BSF shall send 200 OK message ~~-~~and shall supply a transaction identifier to the UE to indicate that the ~~success of the~~ authentication was successful, or that the TID in step 1 was valid.

~~8~~11. The key material Ks is generated in ~~both BSF and~~ UE by concatenating CK and IK. The Ks is used for securing the Ua interface.

Editor's note: The key material Ks is 256 bits long. It is up each NAF to make the usage of the key material specifically.

~~9. BSF may supply a transaction identifier to UE in the cause of Ub interface.~~

4.3.2 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.

NAF starts communication over Zn interface with BSF

- If the NAF already has the 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 due to the fact that the key material has been pushed to the NAF by the BSF, NAF will continue with the protocol used over Ua interface with UE.
- 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 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.
- The NAF derives the keys required to protect the protocol used over Ua interface from the key material in the same way as the UE did.

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. It also does not contain the unsolicited push mechanism from BSF to NAF.

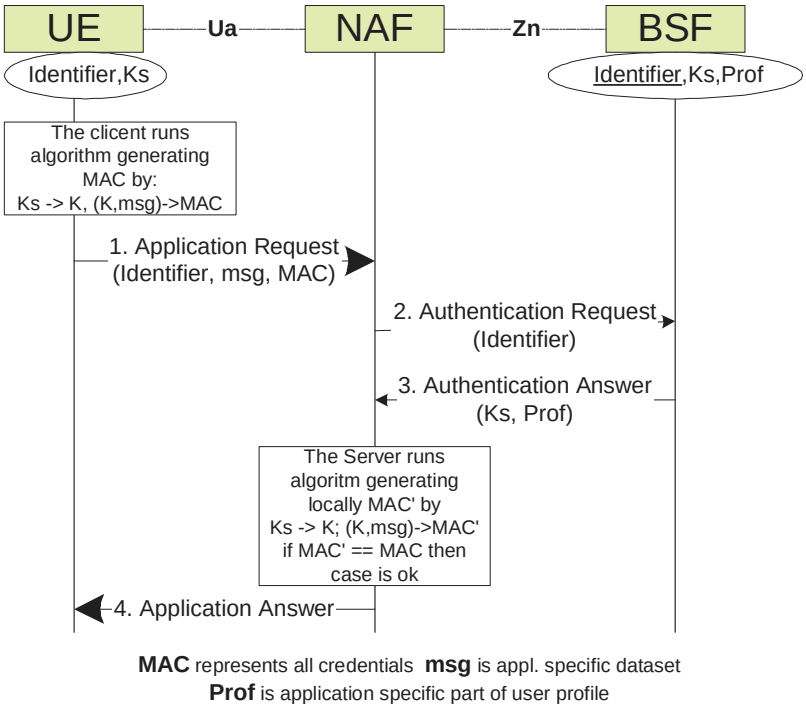


Figure 5: The bootstrapping usage procedure

******* END CHANGE *******