3GPP TSG SA WG3 Security **S3-040342**

10th – 14th May, 2004

Beijing, China

Agenda Item:

Source: Ericsson

Title: Authenticated GBA transaction identifier

Document for: Discussion/Decision

1. Introduction

Generic Bootstrapping Architecture (GBA) has mainly been built under an assumption that Network Authentication Function (NAF) is located in the home network. However, there is already a use case when NAF may be located in visited network, i.e. MBMS. This increases the pressure for SA3 to decide how to implement GBA in roaming scenarios.

This document does not directly propose any GBA roaming model. However, document proposes enhancement to GBA transaction identifier in order to make it more secure also for roaming uses cases. It is proposed that SA3 should consider the proposed transaction identifier format for GBA.

Change to the procedure by which the transaction identifier is created is also proposed. Instead of generating the transaction identifier in BSF, and sending it to UE, it is proposed that UE generates the identifier from the parameters that it already possesses.

2. Problem statement

One problem related to current GBA is that BSF (in the home network) cannot be sure that NAF, who is requesting keying material, is really talking to the UE. The only verification BSF does it based on the validity of transaction identifier. If transaction identifier exists, then BSF will return keying material to NAF.

Anybody who is able to monitor UE-BSF interaction is also able to construct valid transaction identifier (currently specified as RAND@BSF_servers_domain_name). This is not a problem if only authorized NAFs have access to BSF, or if all NAFs and BSFs use some underlying security mechanism such as TLS or IPsec. However, if the 3G networks are more open in the future, the current GBA may not be secure enough. Especially, BSF would benefit from knowing that UE is really involved with NAF at the time NAF makes the key material request.

Also, the current GBA documents define that BSF should generate the transaction identifier, and send it to UE in 200 OK message. This looks like a waste of resources because UE knows all parameters related to the transaction identifier, i.e. RAND and BSF server domain name, as they are currently defined.

2. Solution

This document proposes enhancement to GBA transaction identifier. Instead of using RAND alone as the “username” in transaction identifier, SA3 should consider the following:

- Including AKA session key (CK) to the identifier. In order to avoid revealing the session key to unauthorized parties, a one-way hash function should be used. Binary output form a hash function should be encoded to text format, e.g. by using base64 encoding.
- In order to be able to use the same keying material for other purposes, the key should be concatenated with a static string, the so-called key mask, before using as an input to the hash function.

Example of more secure transaction identifier would be:

Base64_encoded[hash(key-mask | CK)]@BSF_servers_domain_name

where key-mask is a static string such as “3GPP-bootstrapping”

The shortcoming of this transaction identifier is still that it can be used only once. If communication with a NAF is not successful, the same transaction identifier cannot be re-used securely with other NAFs – because the transaction identifier may have been revealed to unauthorized parties. In order to allow re-use of the same transaction identifier with several NAFs securely, the derivation should include a counter value. The counter value could be generated by the UE, and included both in the hash function and as a clear text in the transaction identifier.

Examples of more secure transaction identifier that is not restricted to one use only:

counter.Base64_encoded[hash(counter | key-mask | CK)]@ BSF_servers_domain_name

Note that the transaction identifier should also include a clear text parameter that will identify the subscription and/or end-user public identity. Identification information is needed in BSF in order to find the right password effectively. It may also be used for indicating under which public identity the UE is authenticated. If clear text identification information is not included, the implementation becomes complex. For example, the identifier may include base64 encoded RAND value – in the same way that it is currently specified:

counter.Base64_encoded[hash(counter | key-mask | CK)].Base64_encoded[RAND]@ BSF_servers_domain_name

3. Conclusions

This document proposed an enhancement to transaction identifier in order to further secure GBA infrastructure also in roaming scenarios. Note also that the solution has potential also for other use, e.g. simple “single-sign-on” solution (each transaction identifier is essentially “a ticket” or one-time password that can be sent in clear text).

It is also proposed that transaction identifier is generated in UE and BSF independently in order to save resources in the Ub interface.

Attached CRs implement these changes to 33.220.

3GPP TSG SA WG3 Security — S3#33 **S3-040xxx**

10th – 14th May, 2004, Beijing, China

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **CRNum** ⌘ **rev** **-** ⌘ Current version: 6.0.0 ⌘

*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 [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Authenticated GBA transaction identifier
Source:	⌘	Ericsson
Work item code:	⌘	SEC1-SC
Date:	⌘	29 April 2004
Category:	⌘	
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:

2	*(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)*

Reason for change:	⌘	In current GBA, BSF cannot be sure that NAF, who is requesting keying material, is really talking to the UE. The only verification BSF does it based on the validity of transaction identifier. If transaction identifier exists, then BSF will return keying material to NAF. Anybody who is able to monitor UE-BSF interaction is also able to construct valid transaction identifier.
Summary of change:	⌘	The concept of Authenticated Transaction Identifier is introduced.
Consequences if not approved:	⌘	BSF is not able to verify that UE is really involved when NAF request key material.

Clauses affected:	⌘	4.5.3, A.2

Other specs affected: ⌘	Y	N	
	Y		Other core specifications ⌘
		N	Test specifications
		N	O&M Specifications

Other comments: ⌘

***** Begin of Change ****

4.5.3 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 (i.e. if a key Ks_NAF for the corresponding key derivation parameter NAF_Id_n 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 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 is available in the UE, the UE first agrees on a new key Ks with the BSF over the Ub interface, and then proceeds to derive Ks_NAF;
- if the NAF shares a key with the UE, but an update of that key is needed, e.g. because the key's lifetime has expired, it shall send 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 (cf. 4.5.1);
- the UE generates Authentication Prefix to the Transaction Identifier. UE takes a hash over concatenated counter C, a constant string "3GPP-bootstrapping", and session key CK, and generates a dot separated string where the first part is a clear text counter C, and the second part is the result of the hash function, i.e. "C.hash[counter | "3GPP-bootstrapping" | CK].". UE maintains the counter C for each key Ks starting from value 1, and increases the counter value by one (counter + 1) every time it generates a new Authentication Prefix. Authentication Prefix is generated only once for each Ks_NAF;
- the UE supplies Authenticated Transaction Identifier to the NAF, in the form of concatenated Authentication Prefix and ~~a~~ the 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: The UE shall 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 agreed over the Ub interface 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;

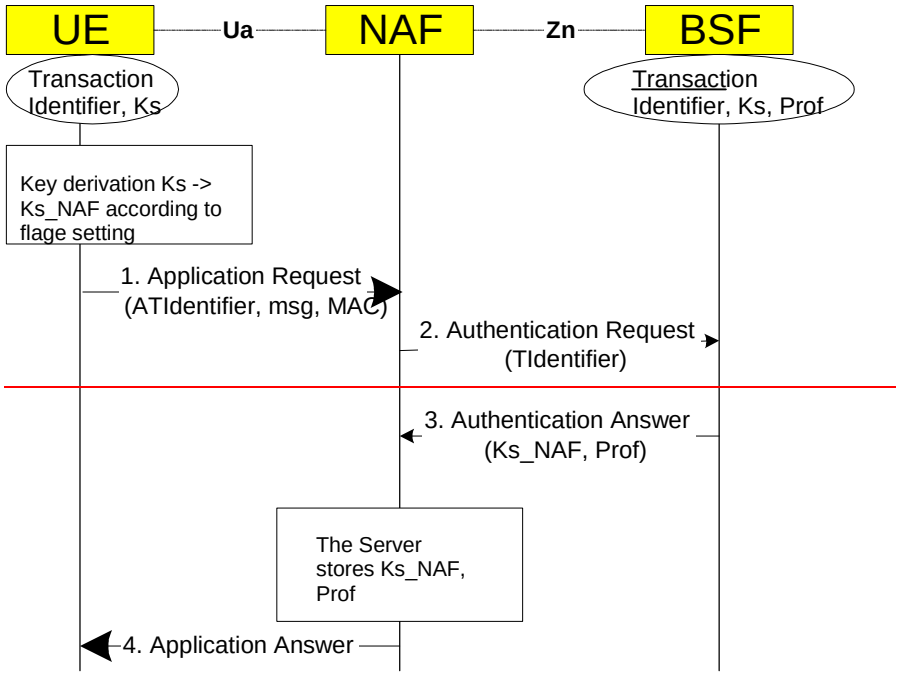
NAF starts communication over Zn interface with BSF

- The NAF requests key material corresponding to Authenticated Transaction Identifier supplied by the UE to the NAF used over Ua interface;
- The BSF checks the validity of Authentication Prefix in the Authenticated Transaction Identifier by re-calculating the hash over concatenated counter C, a constant string "3GPP-bootstrapping", and session key CK, and compares the result with the value received from NAF. BSF maintains the counter C for each key Ks starting from value 1, and stores the last used value every time it receives a new valid authentication prefix. BSF shall not accept counter values that are lower or equal to the last used counter value;
- The BSF derives the keys required to protect the protocol used over Ua interface from the key material Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key material Ks_NAF, as well as the lifetime time of that key material. If the key identified by the Authenticated Transaction Identifier supplied by the NAF is not available at the BSF, or if the Authentication Prefix was not valid, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a key update request to the UE.

NOTE: The NAF shall 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 the Ua interface with the 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.



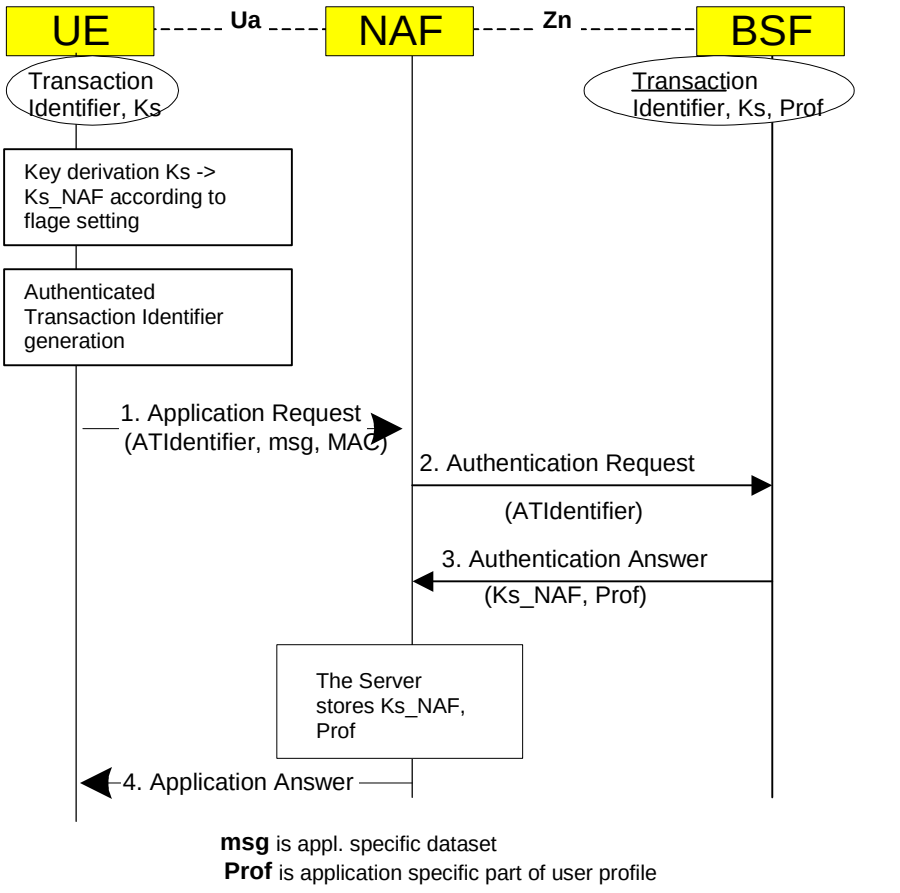


Figure 5: The bootstrapping usage procedure

***** End of Change ****

***** Begin of Change ****

A.2 Generic protocol over Ua interface description

Editor's note: a cross-check with the corresponding stage 3 spec TS 24.cde shall be performed in order to avoid duplication.

The sequence diagram in Figure A.1 describes the generic secure message exchange with HTTP Digest Authentication. The conversation may take place inside a server-authenticated TLS (RFC 2246 [6]) tunnel in which case TLS handshake has taken place before step 1.

In step 1, UE sends an empty HTTP request to a NAF. In step 2, NAF responds with HTTP response code 401 "Unauthorized" which contains a WWW-Authenticate header. The header instructs the UE to use HTTP Digest Authentication with a bootstrapped security association. Quality of protection (qop) attribute is set to "auth-int" meaning that the payload of the following HTTP requests and responses should integrity protected. The realm attribute contains two parts. The first part is a constant string "3GPP-bootstrapping" instructing the UE to use a bootstrapped security association. The second part is the DNS name of the NAF.

In step 3, the UE shall verify that the second part of the realm attribute does in fact correspond to the server it is talking to. In particular, if the conversation is taking place inside a server-authenticated TLS tunnel, the UE shall verify that the server name in the server's TLS certificate matches the server name in the realm attribute of the WWW-Authenticate header. The UE generates client-payload containing the message it wants to send to the server. Then it will generate the HTTP request by calculating the Authorization header values using the Authenticated Transaction Identifier it received from the BSF as username and the session key Ks_NAF as the password, and send the request to NAF in step 4.

When NAF receives the request in step 5, it will verify the Authorization header by fetching the session key Ks_NAF from the bootstrapping server using Zn interface and the Authenticated Transaction Identifier. After successful retrieval, NAF calculates the corresponding digest values using K, and compares the calculated values with the received values in the Authorization header. The NAF shall also verify that the DNS name in the realm attribute matches its own. If the conversation is taking place inside a server-authenticated TLS tunnel, the NAF shall also verify that this DNS name is the same as that of the TLS server. If the verification succeeds, the incoming client-payload request is taken in for further processing. Thereafter, the NAF will generate a HTTP response containing the server-payload it wants to send back to the client in step 6. The NAF may use session key Ks_NAF to integrity protect and authenticate the response.

In step 7, UE receives the response and verifies the Authentication-Info header. If the verification succeeds, the UE can accept the server-payload for further processing.

Additional messages can be exchanged using steps 3 through 7 as many times as is necessary. The following HTTP request and responses shall be constructed according to RFC 261 [3] (e.g., nc parameter shall be incremented by one with each new HTTP request made by UE).

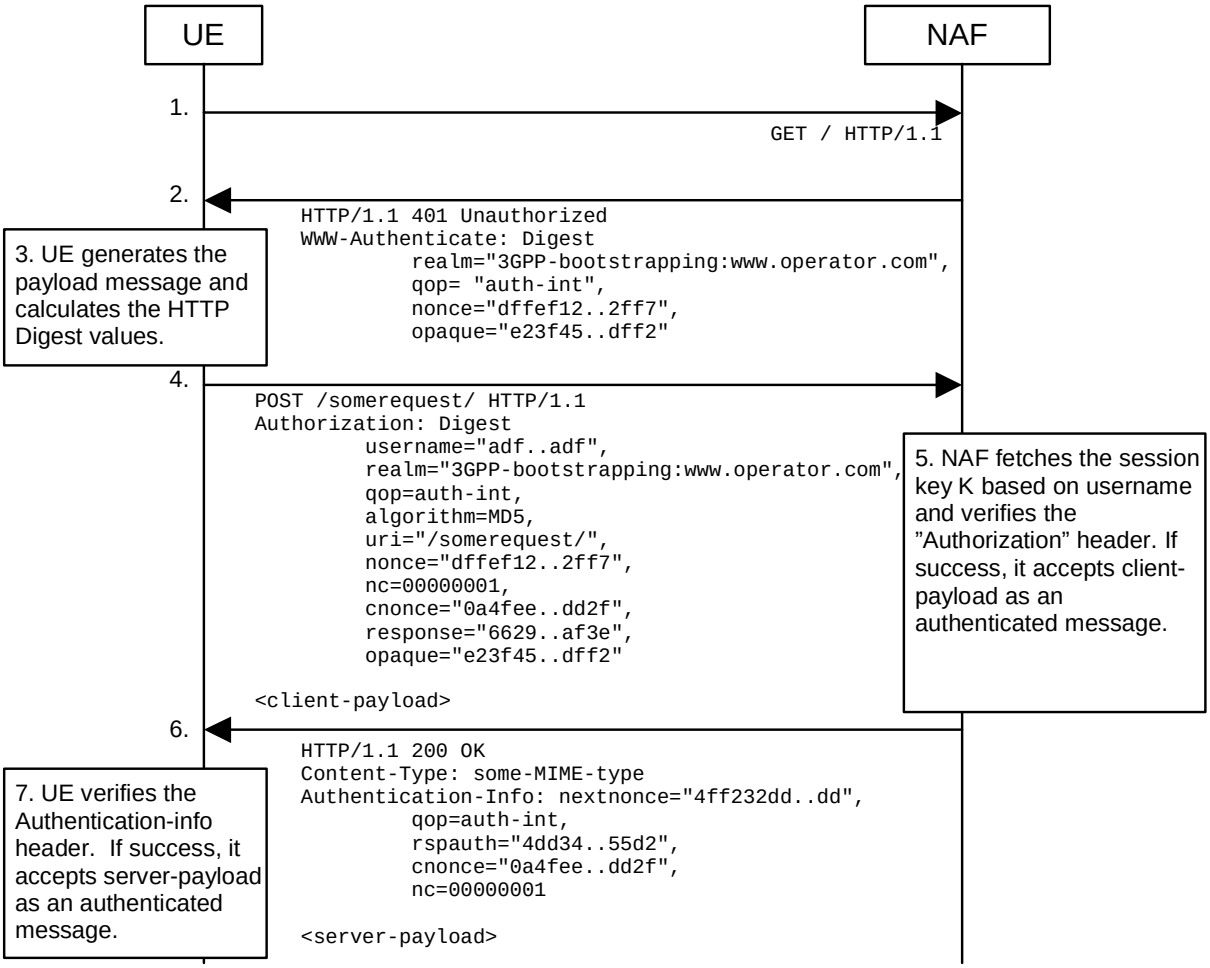


Figure A.1: Generic secure message exchange using HTTP Digest Authentication and bootstrapped security association

***** End of Change ****

3GPP TSG SA WG3 Security — S3#33 **S3-040xxx**

10th – 14th May, 2004, Beijing, China

CR-Form-v7

CHANGE REQUEST

⌘ **33.220** **CR** **CRNum** ⌘ **rev** **-** ⌘ Current version: 6.0.0 ⌘

*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:	⌘	Generation of GBA transaction identifier
Source:	⌘	Ericsson
Work item code:	⌘	SEC1-SC
Date:	⌘	29 April 2004
Category:	⌘	
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:

2	*(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)*

Reason for change:	⌘	Currently, BSF generates the Transaction Identifier, and sends it to UE via the Ub interface. There is no need for this because UE knows all parameters of Transaction Identifier, and consequently is able to generate it by itself.
Summary of change:	⌘	UE generates Transaction Identifier from the parameters it already knows.
Consequences if not approved:	⌘	Procedure is unnecessarily complex.

Clauses affected:	⌘	4.3.4, 4.5.2

Other specs affected:	⌘	Y	N		⌘
		Y		Other core specifications	
			N	Test specifications	
			N	O&M Specifications	

Other comments:	⌘	

***** Begin of Change ****

4.3.4 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 Identifier to the UE.~~

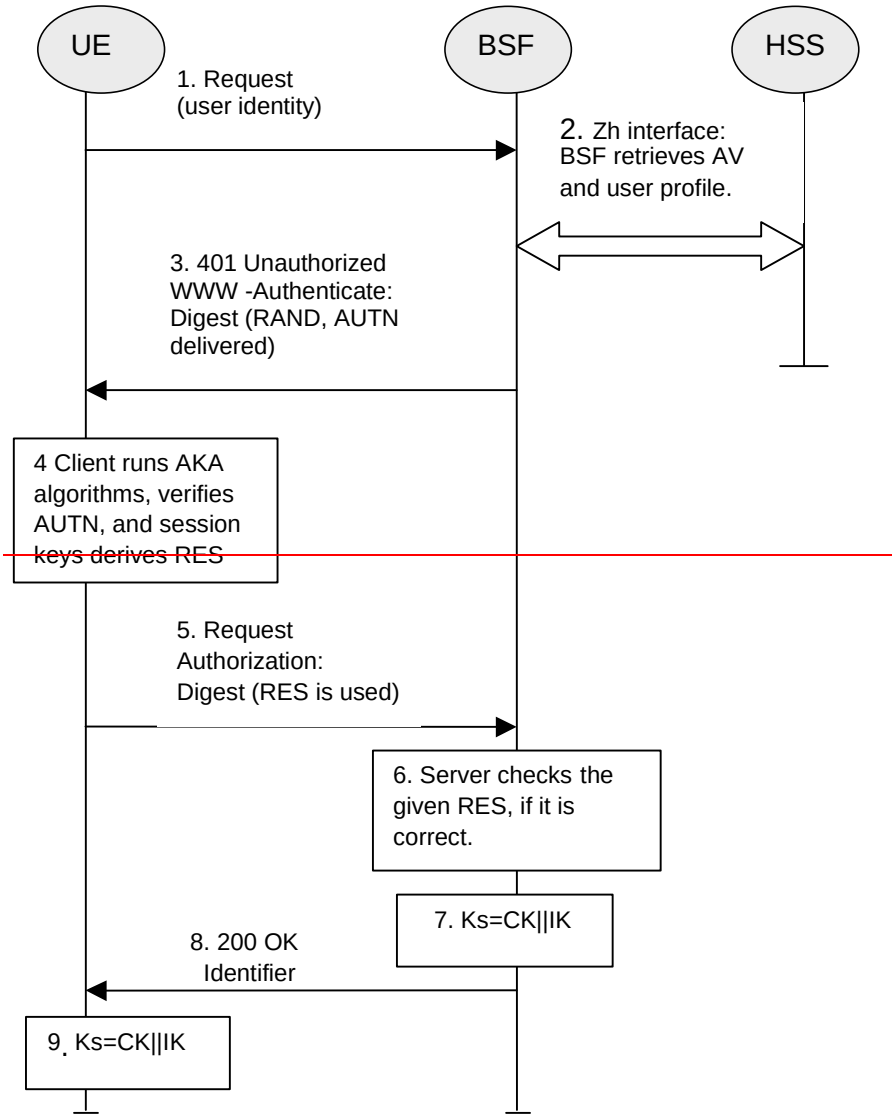
***** End of Change ****

***** Begin of Change ****

4.5.2 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure 3). 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 in UE has expired (cf. subclause 4.5.3).

NOTE: The main steps from the specifications of the AKA protocol in TS 33.102 [2] and the HTTP digest AKA protocol in RFC 3310 [4] are repeated in figure 3 for the convenience of the reader. In case of any potential conflict, the specifications in TS 33.102 [2] and RFC 3310 [4] take precedence.


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. Ks=CK||IK
8. 200 OK
Identifier
9. Ks=CK||IK

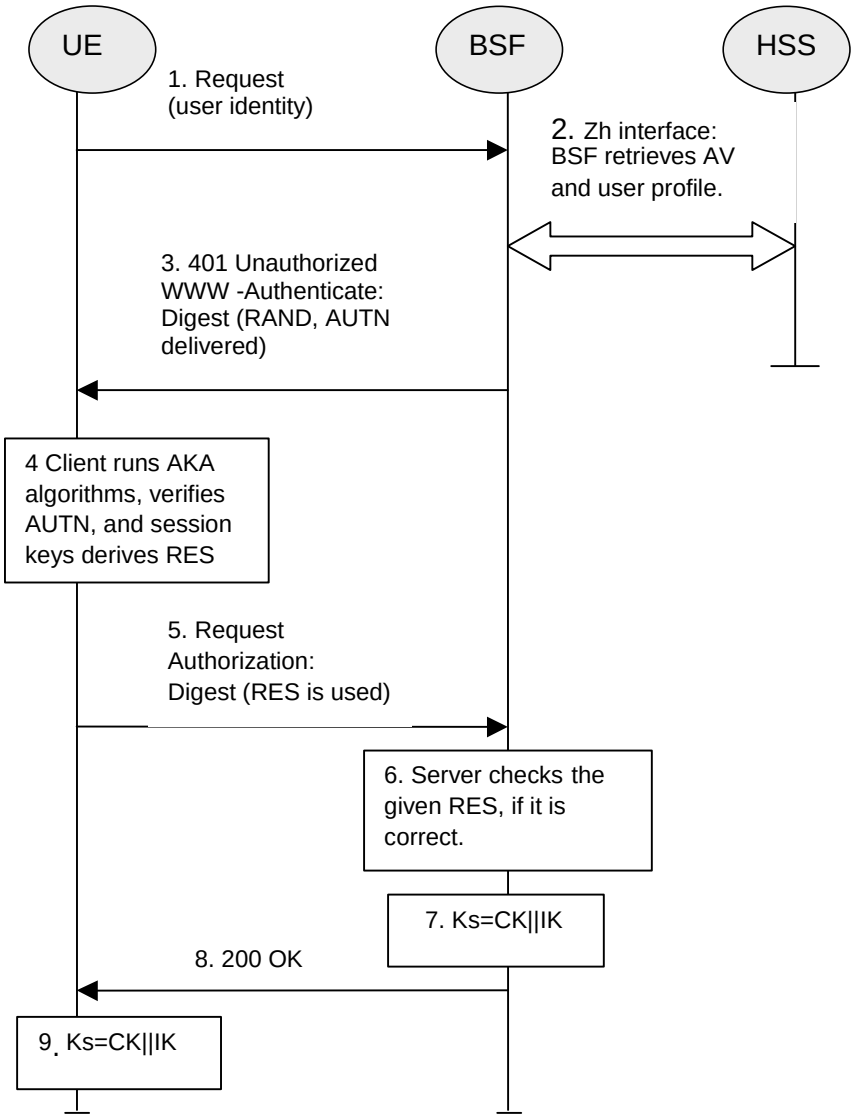


Figure 3: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the user profile and one or a whole batch of Authentication Vectors (AV, AV = RAND||AUTN||XRES||CK||IK) over the 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 checks AUTN to verify that the challenge is from an authorised 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 another HTTP request, containing the Digest AKA response (calculated using RES), to the BSF.
6. The BSF authenticates the UE by verifying the Digest AKA response.
7. The BSF generates key material Ks by concatenating CK and IK. The Transaction Identifier value shall be also generated in format of NAI by taking the RAND value ~~from step 3~~, and the BSF server name from step 3, i.e. RAND@BSF_servers_domain_name.

NOTE: Where exactly the BSF server name is taken, is out of the scope of this specification. Good candidates for BSF name extraction are some parameters available in WWW-Authenticate header, such as “domain” or “realm”, or the DNS name of BSF.

8. The BSF shall send a 200 OK message~~, including a Transaction Identifier,~~ to the UE to indicate the success of the authentication. The BSF also supplies a flag DER_FLAG to the UE, which indicates whether key derivation shall be applied to Ks or not. If key derivation is performed it is to be applied uniformly to all keys shared between any UE and any NAF. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key Ks, and an indication whether multiple key derivation shall be used. The key material Ks is generated in UE by concatenating CK and IK.

9. UE generates the Transaction Identifier by taking the RAND value, and the BSF server name from step 3, and concatenates them into the NAI format, i.e. RAND@BSF_servers_domain_name. Both the UE and the BSF shall use the Ks to derive the key material Ks_NAF, if applicable. 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 and RAND. The NAF_Id consists of the full DNS name of the NAF. KDF shall be implemented in the ME.

Editor's note: The definition of the KDF and the possible inclusion of further key derivation parameters are left to ETSI SAGE and to be included in the Annex B of the present specification.

If multiple key derivation is used then the UE and the BSF store the key Ks with the associated Transaction Identifier for further use, until the lifetime of Ks has expired, or until the key Ks is updated. Otherwise, the key Ks and the Transaction Identifier may be deleted in the UE and in the BSF after the key Ks_NAF has been derived.

***** End of Change ****