Technical Specification Group Services and System Aspects **TSGS#23(04)0175**

Meeting #23, Phoenix, USA, 15 - 18 March 2004

Presentation of Specification to TSG

Source: **SA WG3**

Presentation to: **TSG SA Meeting #23**

Document for presentation: **TS 33.220, Version 2.0.0**

Presented for: **Approval**

Abstract of document:

The Technical specification TS 33.220: Generic Bootstrapping Architecture (GBA) for Release 6 was present in SA#22 (TSGS#22(03) 0583) for information. It specifies a bootstrapping architecture to provide UE a mean to authenticate itself to a Network Application Function (NAF) based on user's subscription. Interfaces Ub, Zh, and Zn are specified in stage-2 level detail. This specification has been further developed during SA3#32, and it is estimated to be 80% ready.

Stage 3 work of the present specification is in progress by CN groups. CN1 is specifying one stage-3 level **TS 24.xxx**, which will specify Ub interface and potentially Ua interface. CN4 is specifying one stage-3 level **TS 29.109**, which specifies Zh and Zn interfaces.

Changes since last presentation to TSG SA:

The following changes were made after TSG SA#22.

- Re-organization of the sections in the specification
- Service discovery function is added
- Lifetime of the key material is included
- New function added in BSF and UE: Key derivation from key material, and multiple key derivation for multiple NAFs.
- Removal of key derivation parameter n
- Using GBA towards the UICC is for FFS
- Protection of the Zn-interface for a NAF within the visited network

Outstanding Issues:

Open issues are listed below. Note that all of the open issues are minor.

- Needed new parameters for subscriber's profile in HSS related to GAA are FFS.
- Method for NAF to indicate that new bootstrapping is required for UE, is dependent on a particular Ua interface and is FFS.
- Whether to retrieve subscriber's profile and whether the identities of the subscriber shall be retrieved via Zn interface, is FFS.
- Key Derivation Function is to be provided by ETSI SAGE.

Contentious Issues:

None.

3GPP TS 33.220 V1.2.1 (2004-3)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (Release 6)

Keywords
Security, GAA

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document describes the security features and a mechanism to bootstrap authentication and key agreement for application security from the 3GPP AKA mechanism. Candidate applications to use this bootstrapping mechanism include but are not restricted to subscriber certificate distribution [5]~~, etc~~. Subscriber certificates support services whose provision mobile operator assists, as well as services that mobile operator provides.

The scope of this specification includes a generic AKA bootstrapping function, an architecture overview and the detailed procedure how to bootstrap the credential.

~~Editor's note~~NOTE: The specification objects are scheduled currently in phases. For this specification release ~~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~~further specification release, other configurations may be considered.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 31.102: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the USIM application".

[2] 3GPP TS 33.102: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture".

[3] Franks J., et al,: "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[4] A. Niemi, et al,: "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC3310, September 2002.

[5] 3GPP TS 33.221: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Support for Subscriber Certificates".

[6] T. Dierks, et al,: "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[7] OMA: “Provisioning Content Version 1.1”, Version 13-Aug-2003. Open Mobile Alliance.

[8] 3GPP TS 23.228: “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 6) ”.

3 Definitions and aAbbreviations

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.

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

Editor's note: Definition to be completed.

Transaction Identifier:

Editor's note: Definition to be completed.

3.2 Abbreviations

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

AK	Anonymity Key
AKA	Authentication and Key Agreement
BSF	Bootstrapping ~~s~~Server ~~f~~Function~~ality~~ ~~BSF is hosted in a network element under the control of an MNO.~~
~~BSP~~	~~BootStrapping Procedure~~
CA	Certificate Authority
~~CMP~~	~~Certificate Management Protocols~~
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
MNO	Mobile ~~n~~Network ~~o~~Operator
NAF	~~Operator controlled n~~Network ~~a~~Application ~~f~~Function ~~functionality.~~ ~~NAF is hosted in a network element under the control of an MNO.~~
~~PKCS~~	~~Public Key Cryptography Standards~~
PKI	Public Key Infrastructure
~~SCP~~	~~Subscriber Certificate Procedure~~
~~UE~~	~~User Equipment~~

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~~establish shared keys. Therefore, 3GPP can provide the "bootstrapping of application security" to authenticate the subscriber by defining a ~~generic~~ Generic ~~bootstrapping~~ Bootstrapping ~~function~~ Architecture (GBA) based on AKA protocol.

4.1 Reference model

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

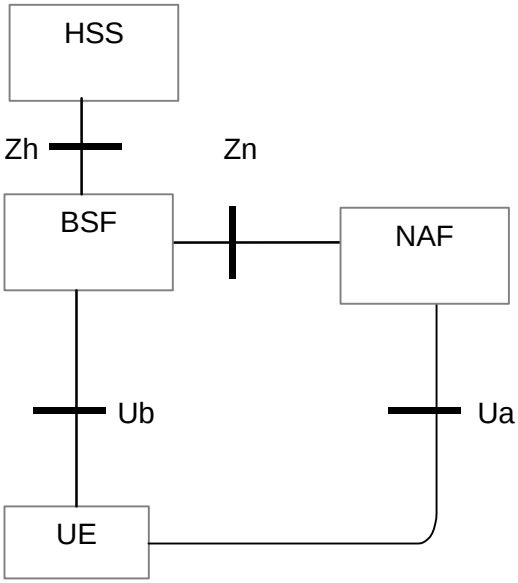


Figure 1: Simple network model for bootstrapping

4.~~1~~2 Network elements

4.~~23.~~2.1 Bootstrapping server function (BSF)

A generic ~~bootstrapping~~ Bootstrapping ~~server~~ Server ~~function~~ 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~~ Network ~~application~~ Application ~~function~~ Function (NAF). ~~The key material must be generated specifically for each NAF independently, that is, for each key uniquely identified by a transaction identifier and that is shared between a UE and a NAF there is a new run of HTTP Digest AKA [4] over the Ub interface.~~ The BSF can restrict the applicability of the key material to a defined set of NAFs by using a suitable key derivation procedure. The generation of key material is specified in section 4.5.2.

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
- Issues with key lifetime are ffs.

4.~~23.~~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~~)~~ are:

- ~~there~~ 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 ~~bootstrapping server function (~~BSF~~)~~ during ~~running~~ the run of the application-specific protocol.
- NAF shall be able to check lifetime of the shared key material.

4.~~23.~~2.3 HSS

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

Editor's note: Needed new subscriber profile parameters are FFS.

4.2.4 UE

~~4.23.2.4 UE~~

The required ~~new~~ functionalities from the 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 (For an example see [5]).

4.3 Requirements and principles for bootstrapping

The following requirements and principles are applicable to bootstrapping procedure:~~Editor's note: The description of AKA bootstrapping shall be added here.~~

- The bootstrapping function shall not depend on the particular ~~network application function~~NAF.
- 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~~NAF needs only to be trusted by the home operator to handle derived key material.
- It shall be possible to support ~~network application functions~~NAF 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.
- ~~Editor's note: The specification objects are scheduled currently in phases. 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.~~
- 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.
- It shall be prevented that a security breach in one ~~application server~~network application functionNAF who using the ~~Generic Bootstrapping Architecture~~GBA, can be used by an attacker to mount successful attacks to the other ~~network application servers functions~~NAFs using the ~~Ggeneric bBootstrapping Aarchitecture~~GBA.

4.~~1~~23.1 Access Independence

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

4.~~1~~23.2 Authentication methods

~~Authentication method that is used to authenticate the bootstrapping function must be dependent on cellular subscription. In other words, a~~Authentication ~~to~~ between the UE and the ~~bootstrapping server function~~BSF shall not be possible without a valid cellular subscription. Authentication shall be based on the 3GPP AKA protocol.

4.~~1~~23.3 Roaming

The roaming subscriber shall be able to utilize the bootstrapping function in the 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~~2~~2~~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~~ Transaction ~~identifier~~ Identifier to the UE.

4.~~1~~2~~2~~3.5 Requirements on Zh interface

The requirements for Zh interface are:

- Mutual authentication, confidentiality and integrity shall be provided.~~The BSF shall be able to communicate securely with the subscriber's HSS.~~

NOTE: This requirement may be fulfilled by physical or proprietary security measures if BSF and HSS are located within the same operator's network. ~~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~~ 3GPP AKA vectors to the BSF in batches.
- The HSS shall be able to send the subscriber's GAA ~~profiles~~ profile information needed for security purposes to the BSF.

~~Editor's note: the intention is not to send all the application specific profile information, but only the information needed for security purposes.~~

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.

Editor's note: This requirement may need to be modified depending on what happens in the case where the profile in the HSS is updated.

~~It is preferred to reuse existing specifications if possible.~~

- The number of different interfaces to HSS should be minimized.

4.~~1~~2~~2~~3.6 Requirements on Zn interface

The requirements for Zn interface are:

- ~~—~~Mutual authentication, confidentiality and integrity shall be provided.

NOTE: This requirement may be fulfilled by physical or proprietary security measures if BSF and NAF are located within the same operator's network.

- The BSF shall verify that the requesting NAF is authorised.
- 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 information needed for security purposes from BSF.
- The BSF shall be able to indicate to the NAF the lifetime of the key material.

Editor's note: Relationship between Transaction Identifier and subscriber identity is ffs. In the case of Presence Ut interface, there are several potential identities that are related to Transaction Identifier, i.e. IMPI and IMPUs.

The subscriber may have several Presence accounts related to same IMPI. Transaction Identifier does not carry enough information on which IMPU the end-user is trying to use.

~~Editor's note: The intention is not to send all the application specific profile information, but only the information needed for security purposes.~~

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

4.3.7 Requirements on Transaction Identifier

Transaction identifier shall be used to bind the subscriber identity to the keying material in Ua, Ub and Zn interfaces.

Requirements for Transaction Identifier are:

- Transaction Identifier shall be globally unique.
- Transaction Identifier shall be usable as a key identifier in protocols used in the Ua interface.
- NAF shall be able to detect the home network and the BSF of the UE from the Transaction Identifier.

Editor's note: Parallel use of GBA and non-GBA infrastructure is ffs. There are use cases when NAF may want to use GBA and non-GBA based infrastructures at the same time. For example, a NAF may want to authenticate subscribers both by using normal HTTP Digest authentication (where the usernames and passwords are distributed using some other mechanism than GBA), and by using GBA based HTTP Digest. However, it seems that in most telecommunication protocols, the server side (i.e. NAF) controls the name space related to key identifiers (cf. Transaction Identifier). For example, in HTTP authentication, the server issues the usernames, and does not allow the re-use of already existing usernames. The parallel use of GBA and non-GBA based infrastructures may cause conflicts on Transaction Identifier namespace. In particular, BSF may assign Transaction Identifier values that NAFs are already using with non-GBA UEs.

Editor's note: GBA shall further specify on how security associations are removed and/or updated in NAF.

4.~~23~~4 Bootstrapping architecture and reference points ~~Bootstrapping architecture~~

4.~~23~~4.1 Ub interface ~~Reference modelProtocols~~

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

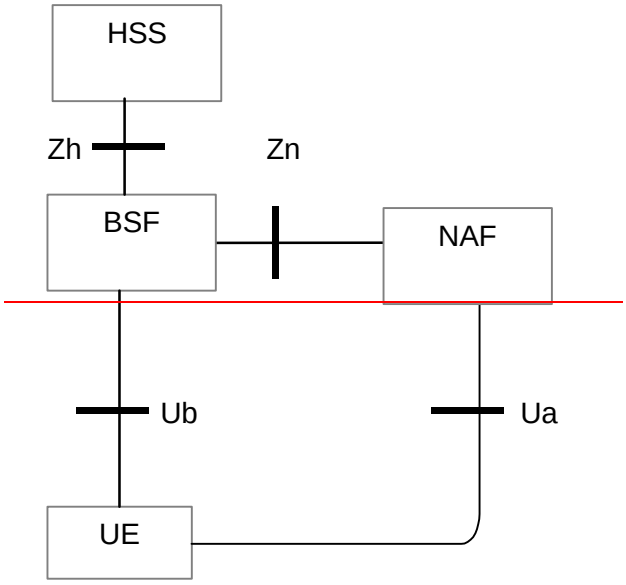


Figure 1: Simple network model for bootstrapping

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.

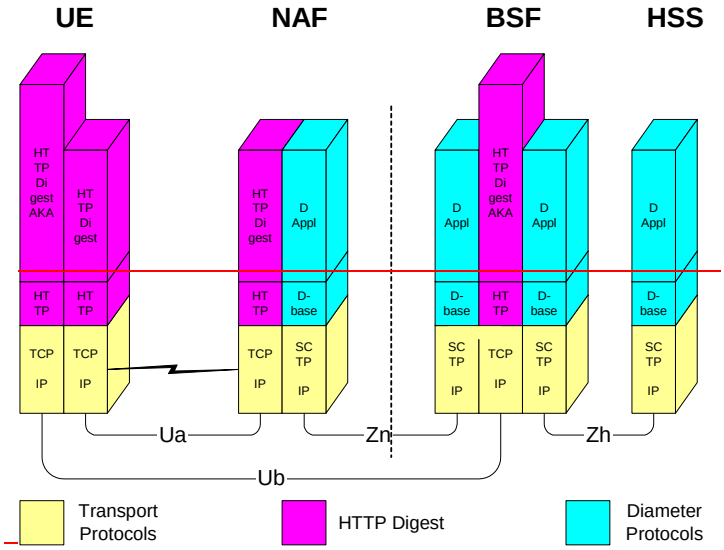


Figure 2: Protocol stack architecture

Editor's note: The protocol on the Ua interface is NAF specific. An example of the Ua interface protocol when the NAF is HTTP based is given in Annex A.

4.23.2 Network elements

4.23.2.1 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, that is, for each key uniquely identified by a transaction identifier and that is shared between a UE and a NAF there is a new run of HTTP Digest

AKA [4] over the Ub interface. The BSF can restrict the applicability of the key material to a defined set of NAFs by using a suitable key derivation procedure.

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
- Issues with key lifetime are ffs.

4.23.2.2 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 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 bootstrapping server function (BSF) during running the run of the application specific protocol.

4.23.2.3 HSS

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

Editor's note: Needed new parameters are FFS.

4.23.2.4 UE

The required new functionalities from the 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]).

4.234.32 Reference points

4.234.32.1 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 note: The solution for CS domain is ffs.

Reference point Ub provides mutual authentication between the UE and the BSF. It allows the UE to bootstrap the session keys based on 3GPP AKA infrastructure.

~~4.234.32.1.1 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 3GPP AKA infrastructure. The session key as result of key agreement functionality, is used to support further applications e.g. certificate issuer.~~

~~4.234.32.1.2 Protocol~~

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

4.234.32.2 Ua interface

The Ua interface carries~~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 the NAF would be the PKI portal.

4.234.32.3 Zh interface

Zh interface ~~is~~ protocol used between the BSF and the HSS ~~to~~ allows 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.

4.234.32.4 Zn interface

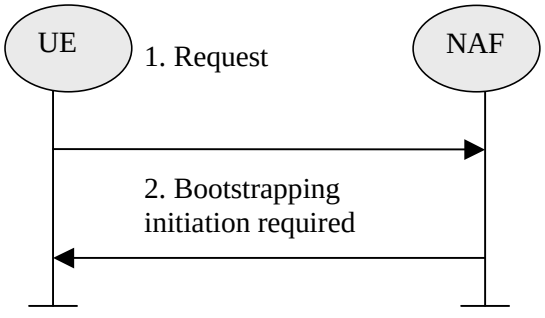
Zn interface is used by the NAF to fetch the key material agreed during a previous HTTP Digest AKA protocol run over Ub interface from the BSF. It may also be used to fetch subscriber profile information from the BSF.

4.345 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.

4.345.1 Initiation of bootstrapping

When a UE wants to interact with a~~n~~ NAF, but it does not know if the bootstrapping procedure is required, it shall contact the NAF for further instructions (see ~~figure~~ Figure ~~3~~2).



~~Figure 3~~Figure 2: Initiation of bootstrapping

1. UE starts communication over Ua interface with the NAF without any bootstrapping -related parameters.

2. If the NAF requires 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 specified in the relevant stage 3-specifications.ffs.Editor's note: 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).

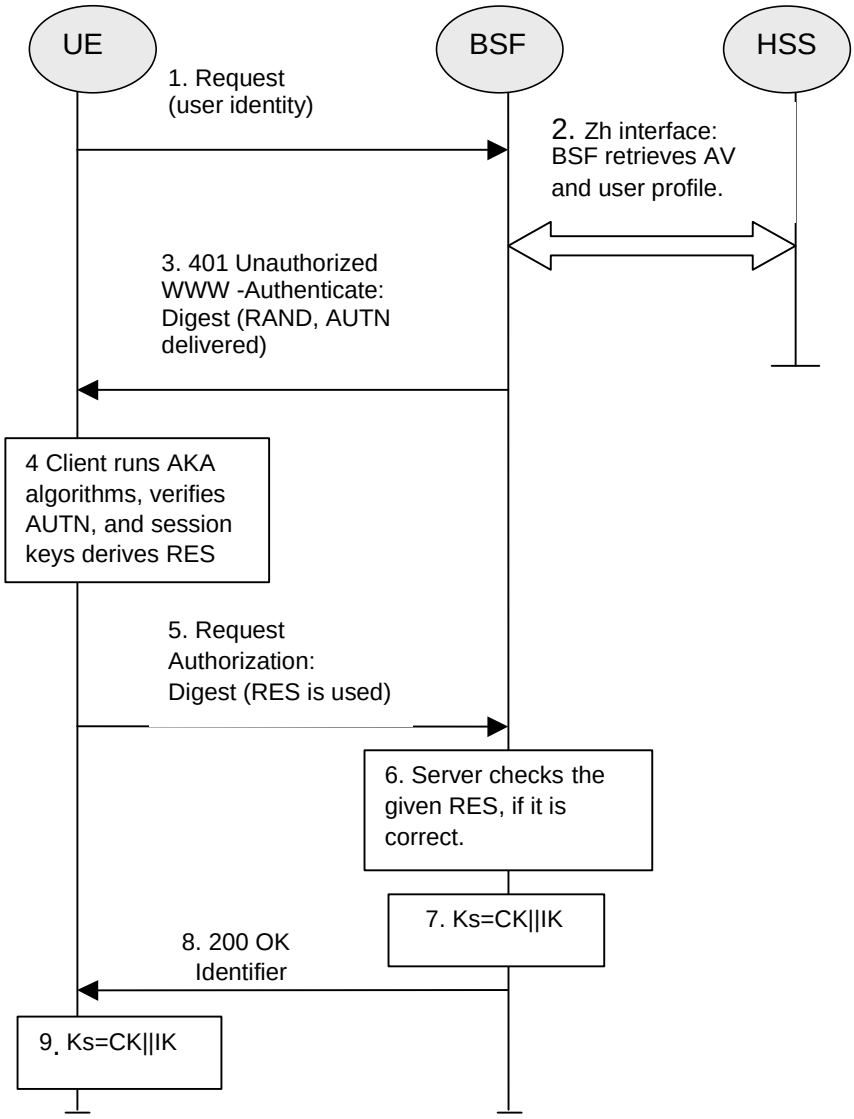
4.345.2 Bootstrapping procedures

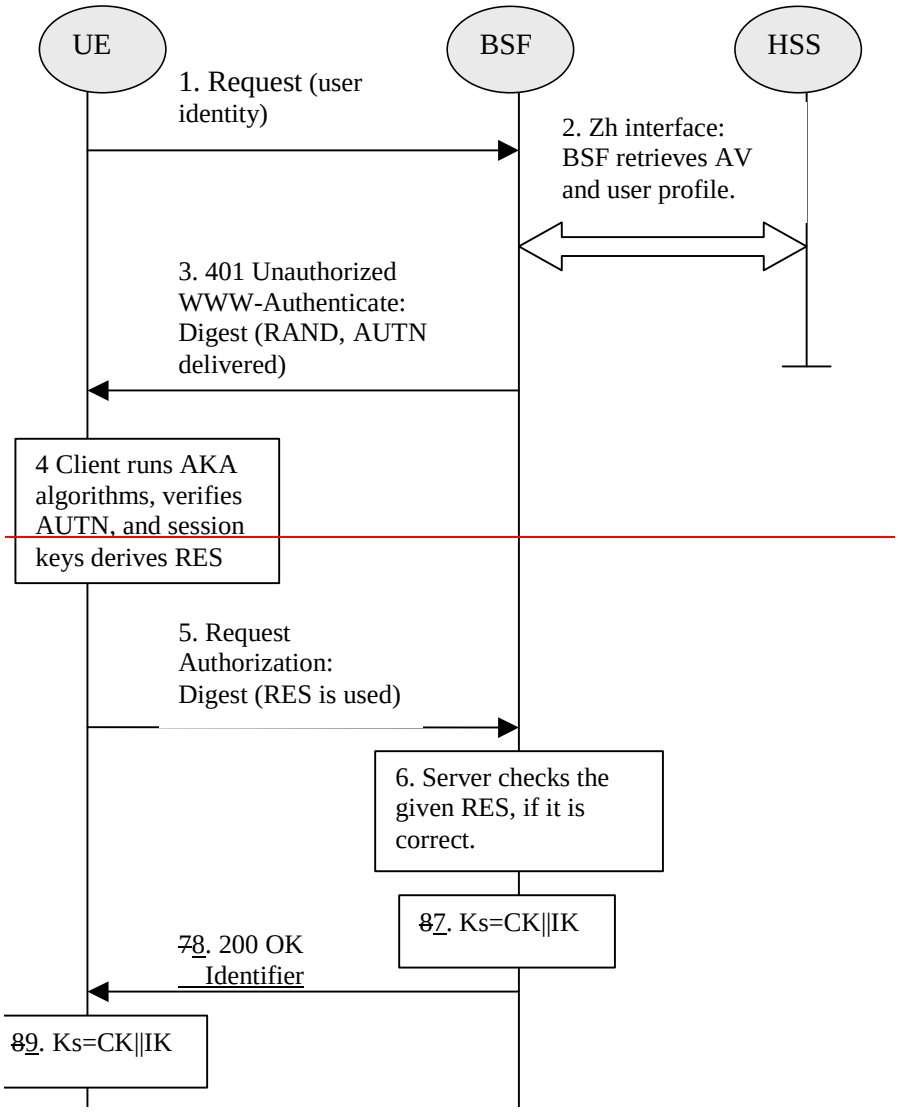
When a UE wants to interact with an NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see fFigure 43). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message 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 [2] and the HTTP digest AKA protocol in [4] are repeated in Figure 3 for the convenience of the reader. In case of any potential conflict, the specifications in [2] and [4] take precedence.

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 (cf. subclause 4.3.3).


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 4~~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~~a challenge, i.e. the~~ Authentication ~~Vector~~ 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 ~~calculates the message authentication code (MAC) so as~~ checks AUTN to verify that the challenge is from an auth~~orised~~~~enticated~~ 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~~, again, with~~containing the Digest AKA response (calculated using RES)~~, as the response~~ to the BSF.

6. The BSF authenticates the UE by verifying the Digest AKA response.~~If the RES equals to the XRES that is in the AV, the UE is authenticated.~~

7. The 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.~~ 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, i.e. RAND@BSF_servers_domain_name.

8. The BSF shall send a 200 OK message, ~~and shall supply~~including a ~~transaction identifier~~Transaction Identifier, to the UE to indicate the success of the authentication. The BSF ~~may~~ also ~~supply~~ supplies a flag DER_FLAG to the UE, which indicates whether key derivation shall be applied to Ks or not. ~~the parameter *n* used to determine the NAF_Id_n (cf. previous next 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.~~ 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.

~~9.~~ The key material Ks is generated in UE by concatenating CK and IK.

9. ~~The~~ Both the UE and the BSF shall use the Ks ~~is used~~ 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 ~~–n~~ and RAND. The NAF_Id ~~–n~~ consists of the ~~n rightmost domain labels in the~~full 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.~~. KDF shall be implemented in the ME.

~~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".~~

~~If the parameter *n* is not supplied then no key derivation is performed, i.e. Ks = Ks_NAF.~~

Editor's note: The definition of the KDF and the possible inclusion of further key derivation parameters ~~is~~ 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.

4.~~3~~45.3 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), ~~there is no need for NAF to retrieve the key(s) over Zn interface~~, 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~~is necessary~~, e.g. because the key's lifetime has expired, it shall send~~s~~ 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~~ (cf. 4.5.1).

- ~~It is assumed t~~Th~~at~~e UE supplies ~~sufficient information~~Transaction Identifier ~~–~~to the NAF, in the form of~~e.g.~~ a ~~transaction~~ Transaction ~~identifier~~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~~ shall adapt ~~–~~the key material Ks_NAF to ~~the~~ 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 ~~the information~~Transaction Identifier supplied by the UE to the NAF ~~(in the form of~~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 Ks and the key derivation parameters, as specified in clause 4.~~3~~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 ~~transaction~~ Transaction ~~identifier~~ 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~~ 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.

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

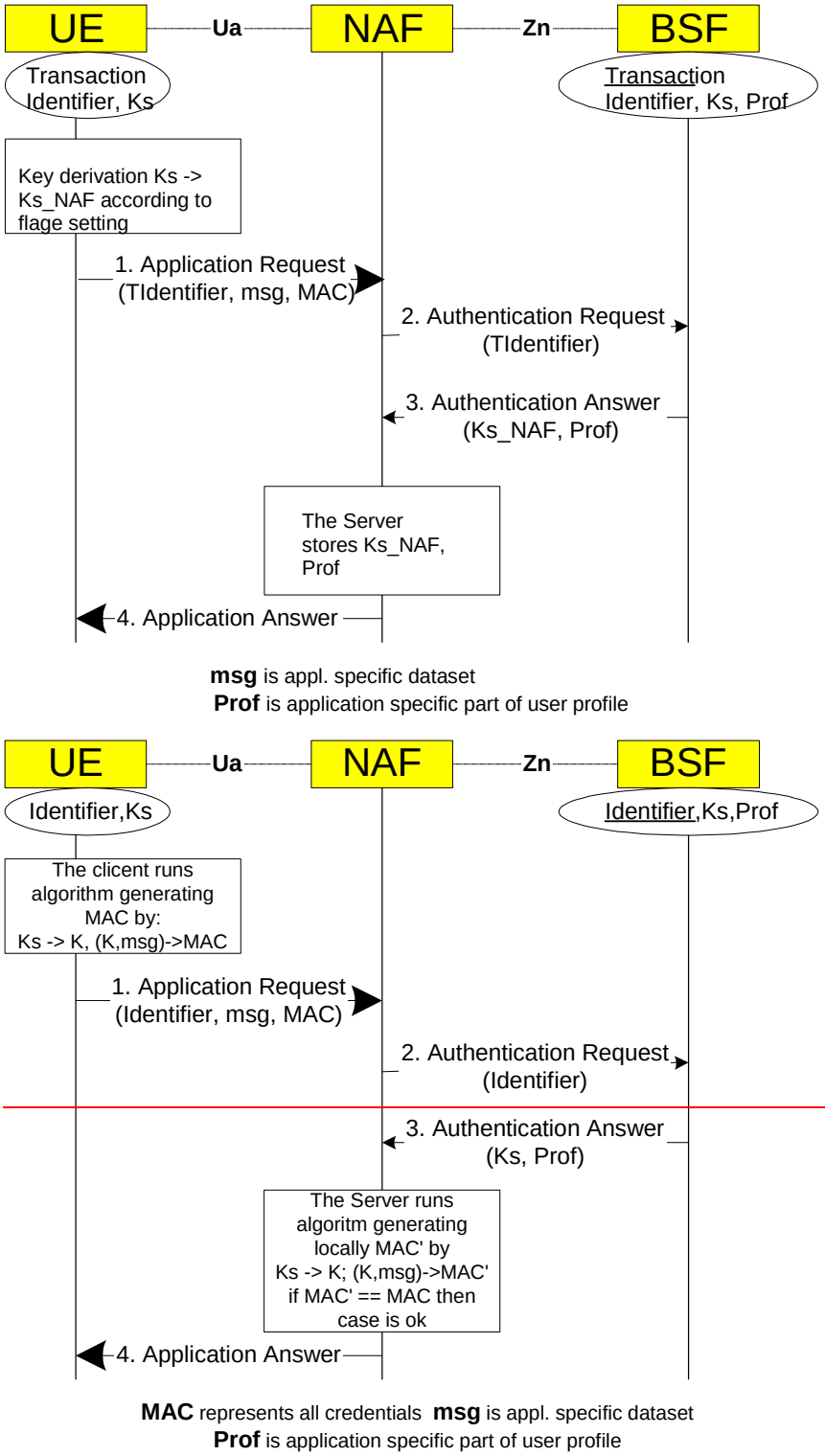


Figure 5: The bootstrapping usage procedure

4.5.4 Procedure related to service discovery

To enable the bootstrapping procedure, a procedure needs to be described on how to discover the location of BSF. It shall be possible to enable the terminal to be configured either manually or automatically via one of the following approaches:

- The address information shall be published via reliable channel. Subscribers shall store all the parameters as part of the initial establishment of IP connectivity. The addresses need to be input only once.

- The address information shall be pushed automatically to the UE over the air interface when the subscription to bootstrapping service is accepted. All the parameters shall be saved in the UE and used the same manner as above. The procedure is specified in [7].

- The location information shall be discovered automatically based on DHCP, after the IP connectivity has been established. The DHCP server shall provide the UE with the domain name of a BSF and the address of a Domain Name Server (DNS) that is capable of resolving the Fully Qualified Domain Name (FQDN) of the BSF. The procedure is specified in [8].

NOTE: The location of DHCP server may be pushed to UE through the procedure specified in [7].

Annex A (informative): Generic secure message exchange using HTTP Digest Authentication

A.1 Introduction

~~Editor's note:~~ This annex describes how HTTP Digest Authentication can be used between UE and any NAF where the protocol over Ua interface is based on HTTP messaging. ~~The protocol over Ua interface may depend upon the final choice of scheme made by SA WG3 and this will need to be reviewed later by SA WG3.~~

HTTP Digest Authentication ~~model~~ can also be used as a generic authentication and integrity protection method towards any new NAF The Generic Bootstrapping Architecture specified in this document enables the NAF and the UE ~~.~~ ~~If a new NAF uses BSF based security association, it could use this generic method~~ to mutually authenticate ~~the UE (and UE authenticate the NAF)~~each other and integrity protect any payload being transferred between NAF and UE. ~~–~~ As a generic method, it will speed up the specification of new NAFs since the authentication and message integrity protection part of Ua interface are taken care of by HTTP Digest Authentication. It will also ease the implementation of GBA~~BSF~~ based authentication in NAFs because there would be one well-defined way to do it.

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~~Figure A.1 describes the generic secure message exchange with HTTP Digest Authentication. The conversation may take place inside a server-authenticated TLS [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 ~~transaction~~ Transaction ~~i~~Identifier ~~(base64 encoded)~~ it received from the BSF as username and the session key Ks_NAF ~~(base64 encoded)~~ 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 ~~transaction~~ Transaction ~~identifier~~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 ~~must~~ shall be constructed according to [3] (e.g., nc parameter ~~must~~ shall be incremented by one with each new HTTP request made by UE).

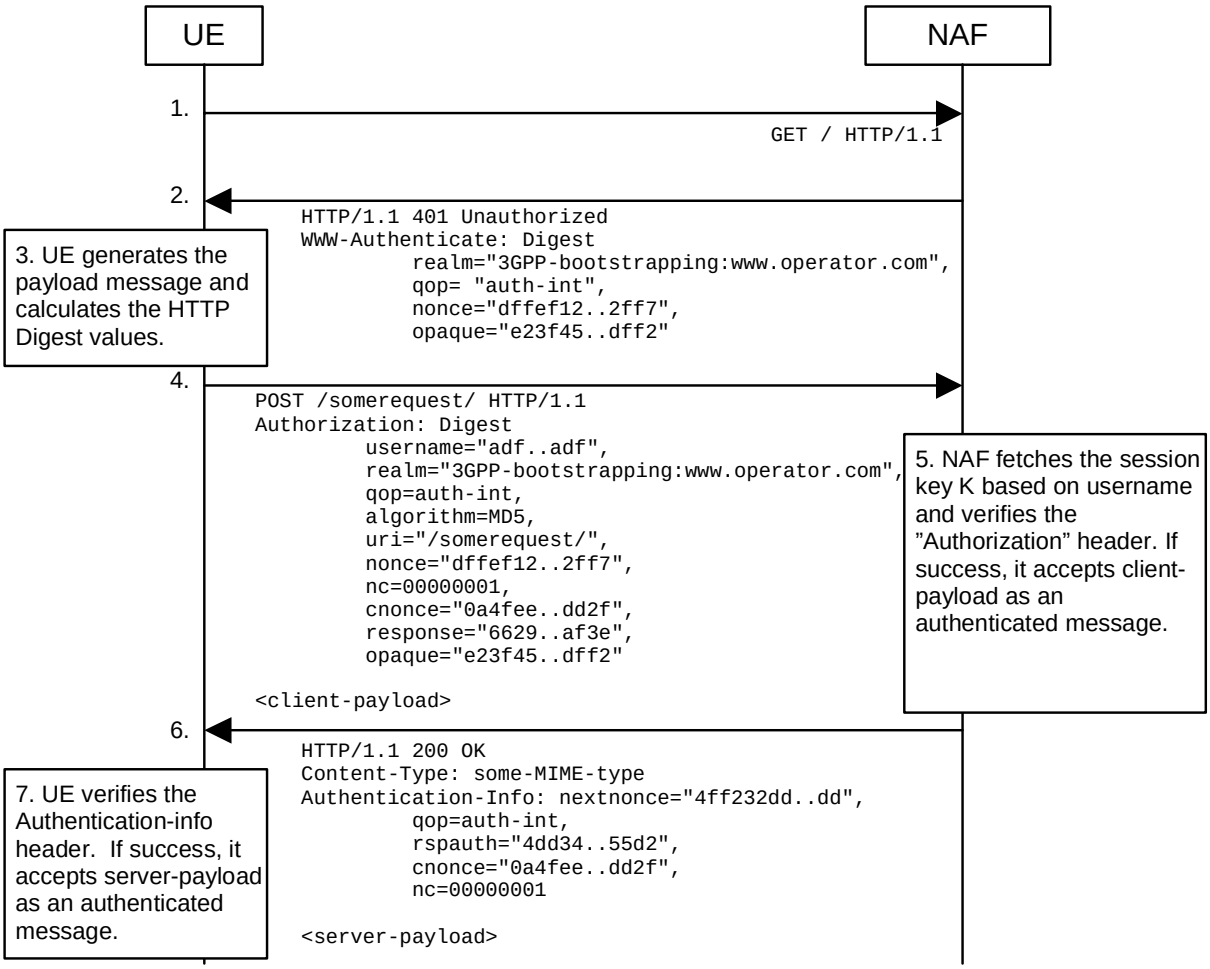


Figure A.1: Generic secure message exchange using HTTP Digest Authentication and bootstrapped security association

Annex B (normative): Specification of the key derivation function KDF

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

Annex ~~B~~C (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2003-10	SA3#30				New draft TS: Generic Bootstrapping Architecture (GBA). Extracted from 33.109 clause 4 and Annex B.		0.1.0
2003-10	SA3#30	S3-030537			New interface names.		0.1.0
2003-10	SA3#30	S3-030538			Requirements for Ub and Zh interfaces added.	0.1.0	0.1.1
2003-10	SA3#30	S3-030545			NAF initiated bootstrapping added	0.1.0	0.1.1
2003-10					Imported Zn interface requirements from SSC TS.	0.1.0	0.1.1
2003-11	SA3#31	S3-030728			Bootstrapping procedure: merging of last two messages	0.1.1	0.2.0
2003-11	SA3#31	S3-030793			Key separation	0.1.1	0.2.0
2003-11	SA3#31	S3-030794			Removal of application specific user profile requirements from GBA	0.1.1	0.2.0
2003-11					Annex A: changed "session key K" to "session key Ks"	0.1.1	0.2.0
2003-12	SP-22	SP-030583	-	-	Presentation to TSG SA#22 for Information	0.2.0	1.0.0
2004-02	SA3#32	S3-040189			Service discovery is added for review of other group	1.0.0	1.1.0
2004-02	SA3#32	S3-040060			Editorial changes throughout the specification	1.1.0	1.2.0
2004-02	SA3#32	S3-040024			Editorial changes throughout the specification	1.1.0	1.2.0
2004-02	SA3#32	S3-040078			Editorial changes on removal of unnecessary texts	1.1.0	1.2.0
2004-02	SA3#32	S3-040157			GBA Transaction Identifier requirements	1.1.0	1.2.0
2004-02	SA3#32	S3-040191			Life time of the bootstrapping information	1.1.0	1.2.0
2004-02	SA3#32	S3-040154			Deletion of parameter n	1.1.0	1.2.0
2004-02	SA3#32	S3-040041			Key handling in the UE in a Generic Bootstrapping Architecture	1.1.0	1.2.0
2004-02	SA3#32	S3-040161			Multiple key derivation in a Generic Bootstrapping Architecture	1.1.0	1.2.0
2004-02					Editorial update for submission to TSG SA for approval	1.2.0	1.2.1

3GPP TS 33.220 V2.0.0 (2004-03)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (Release 6)

Keywords
Security, GAA

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document describes the security features and a mechanism to bootstrap authentication and key agreement for application security from the 3GPP AKA mechanism. Candidate applications to use this bootstrapping mechanism include but are not restricted to subscriber certificate distribution TS 33.221 [5]. Subscriber certificates support services whose provision mobile operator assists, as well as services that mobile operator provides.

The scope of this specification includes a generic AKA bootstrapping function, an architecture overview and the detailed procedure how to bootstrap the credential.

NOTE: The specification objects are scheduled currently in phases. For this specification release, only the case is considered where bootstrapping server functionality and network application function are located in the same network as the HSS. In further specification release, other configurations may be considered.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document.*

[1] 3GPP TS 31.102: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the USIM application".

[2] 3GPP TS 33.102: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security architecture".

[3] Franks J., et al,: "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[4] A. Niemi, et al,: "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC 3310, September 2002.

[5] 3GPP TS 33.221: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Support for Subscriber Certificates".

[6] T. Dierks, et al,: "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[7] OMA: “Provisioning Content Version 1.1”, Version 13-Aug-2003. Open Mobile Alliance.

[8] 3GPP TS 23.228: “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; IP Multimedia Subsystem (IMS); Stage 2 (Release 6) ”.

3 Definitions and abbreviations

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.

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

Editor's note: Definition to be completed.

Transaction Identifier:

Editor's note: Definition to be completed.

3.2 Abbreviations

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

AK	Anonymity Key
AKA	Authentication and Key Agreement
BSF	Bootstrapping Server Function
CA	Certificate Authority
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HSS	Home Subscriber System
IK	Integrity Key
KDF	Key Derivation Function
MNO	Mobile Network Operator
NAF	Network Application Function
PKI	Public Key Infrastructure

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 establish shared keys. Therefore, 3GPP can provide the "bootstrapping of application security" to authenticate the subscriber by defining a Generic Bootstrapping Architecture (GBA) based on AKA protocol.

4.1 Reference model

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

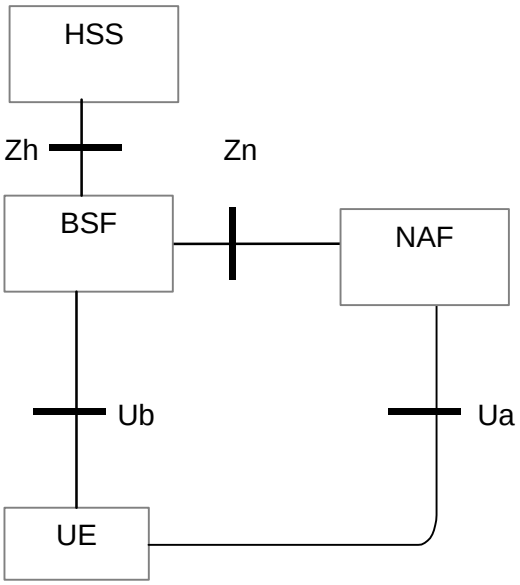


Figure 1: Simple network model for bootstrapping

4.2 Network elements

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 an operator-controlled Network Application Function (NAF). The BSF can restrict the applicability of the key material to a defined set of NAFs by using a suitable key derivation procedure. The generation of key material is specified in section 4.5.2.

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
- Issues with key lifetime are ffs.

4.2.2 Network application function (NAF)

After the bootstrapping has been completed, the UE and an 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 an 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 check lifetime of the shared key material.

4.2.3 HSS

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

Editor's note: Needed new subscriber profile parameters are FFS.

4.2.4 UE

The required functionalities from the 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;
- support of NAF-specific application protocol (For an example see TS 33.221 [5]).

4.3 Requirements and principles for bootstrapping

The following requirements and principles are applicable to bootstrapping procedure:

- the bootstrapping function shall not depend on the particular NAF;
- the server implementing the bootstrapping function needs to be trusted by the home operator to handle authentication vectors;
- the server implementing the NAF needs only to be trusted by the home operator to handle derived key material;
- it shall be possible to support NAF 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;
- it shall be prevented that a security breach in one NAF who using the GBA, can be used by an attacker to mount successful attacks to the other NAFs using the GBA.

4.3.1 Access Independence

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

4.3.2 Authentication methods

Authentication between the UE and the BSF shall not be possible without a valid cellular subscription. Authentication shall be based on the 3GPP AKA protocol.

4.3.3 Roaming

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

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.

4.3.5 Requirements on Zh interface

The requirements for Zh interface are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE: This requirement may be fulfilled by physical or proprietary security measures if BSF and HSS are located within the same operator's network.

- the BSF shall be able to send bootstrapping information request concerning a subscriber;
- the HSS shall be able to send 3GPP AKA vectors to the BSF in batches;
- the HSS shall be able to send the subscriber's GAA profile information needed for security purposes 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;

Editor's note: This requirement may need to be modified depending on what happens in the case where the profile in the HSS is updated.

- the number of different interfaces to HSS should be minimized.

4.3.6 Requirements on Zn interface

The requirements for Zn interface are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE: This requirement may be fulfilled by physical or proprietary security measures if BSF and NAF are located within the same operator's network.

- The BSF shall verify that the requesting NAF is authorised;
- 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 information needed for security purposes from BSF;
- The BSF shall be able to indicate to the NAF the lifetime of the key material.

Editor's note: Relationship between Transaction Identifier and subscriber identity is ffs. In the case of Presence Ut interface, there are several potential identities that are related to Transaction Identifier, i.e. IMPI and IMPUs. The subscriber may have several Presence accounts related to same IMPI. Transaction Identifier does not carry enough information on which IMPU the end-user is trying to use.

4.3.7 Requirements on Transaction Identifier

Transaction identifier shall be used to bind the subscriber identity to the keying material in Ua, Ub and Zn interfaces.

Requirements for Transaction Identifier are:

- Transaction Identifier shall be globally unique;
- Transaction Identifier shall be usable as a key identifier in protocols used in the Ua interface;
- NAF shall be able to detect the home network and the BSF of the UE from the Transaction Identifier.

Editor's note: Parallel use of GBA and non-GBA infrastructure is ffs. There are use cases when NAF may want to use GBA and non-GBA based infrastructures at the same time. For example, a NAF may want to authenticate subscribers both by using normal HTTP Digest authentication (where the usernames and passwords are distributed using some other mechanism than GBA), and by using GBA based HTTP Digest. However, it seems that in most telecommunication protocols, the server side (i.e. NAF) controls the name space related to key identifiers (cf. Transaction Identifier). For example, in HTTP authentication, the server issues the usernames, and does not allow the re-use of already existing usernames. The parallel use of GBA and non-GBA based infrastructures may cause conflicts on Transaction Identifier namespace. In particular, BSF may assign Transaction Identifier values that NAFs are already using with non-GBA UEs.

Editor's note: GBA shall further specify on how security associations are removed and/or updated in NAF.

4.4 Bootstrapping architecture and reference points

4.4.1 Ub interface

The reference point Ub is between the UE and the BSF. Reference point Ub provides mutual authentication between the UE and the BSF. It allows the UE to bootstrap the session keys based on 3GPP AKA infrastructure.

The HTTP Digest AKA protocol, which is specified in RFC 3310 [4], is used on the Ub interface. It is based on the 3GPP AKA TS 33.102 [2] protocol. The interface to the USIM is as specified in TS 31.102 [1].

4.4.2 Ua interface

The Ua interface carries 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 TS 33.221 [5], it is a protocol, which allows the user to request certificates from the NAF. In this case the NAF would be the PKI portal.

4.4.3 Zh interface

Zh interface protocol used between the BSF and the HSS allows 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.

4.4.4 Zn interface

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

4.5 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 the key material generation procedure.

4.5.1 Initiation of bootstrapping

When a UE wants to interact with a NAF, but it does not know if the bootstrapping procedure is required, it shall contact the NAF for further instructions (see figure 2).

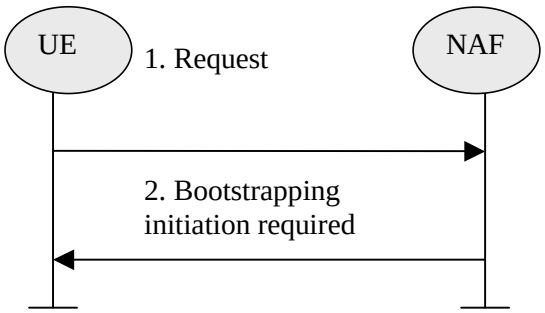


Figure 2: Initiation of bootstrapping

1. UE starts communication over Ua interface with the NAF without any bootstrapping-related parameters.
2. If the NAF requires 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 specified in the relevant stage 3-specifications.

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.

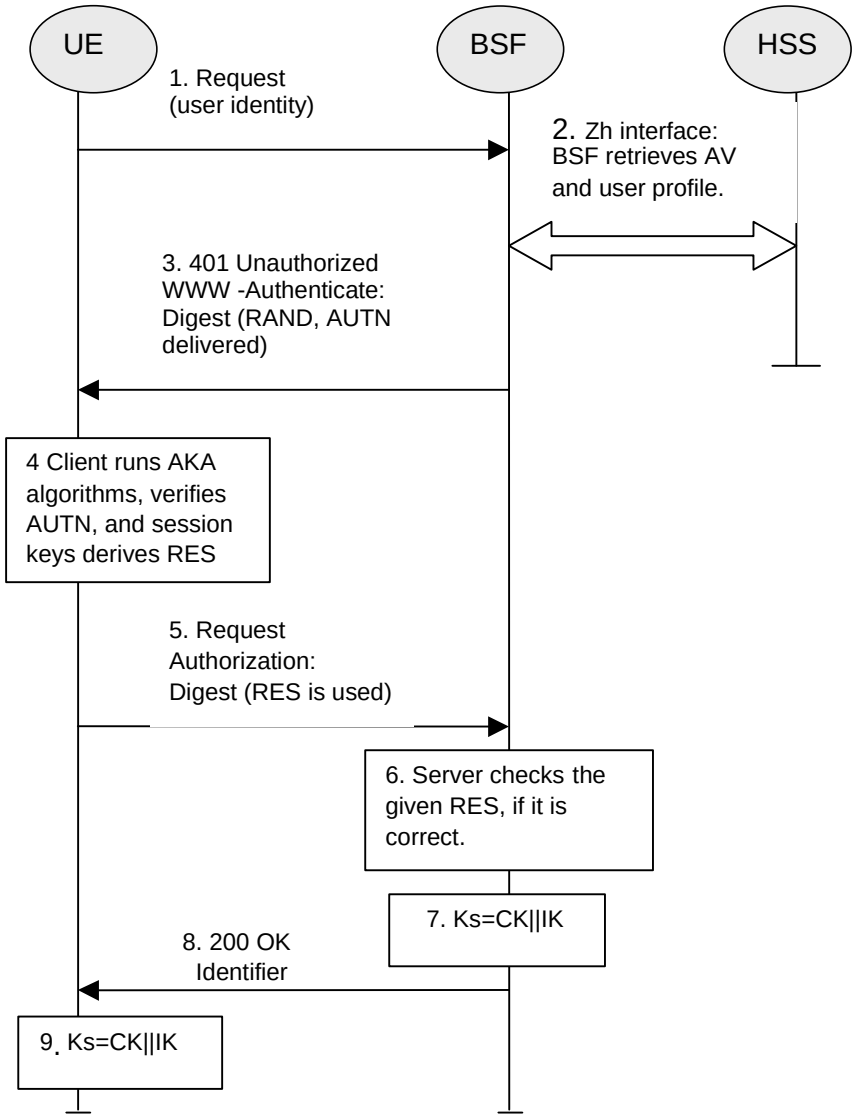


Figure 3: The bootstrapping procedure

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, i.e. RAND@BSF_servers_domain_name.

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. 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.

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 supplies Transaction Identifier to the NAF, in the form of 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: 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 Transaction Identifier supplied by the UE to the NAF used over Ua interface;
- 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 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: 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.

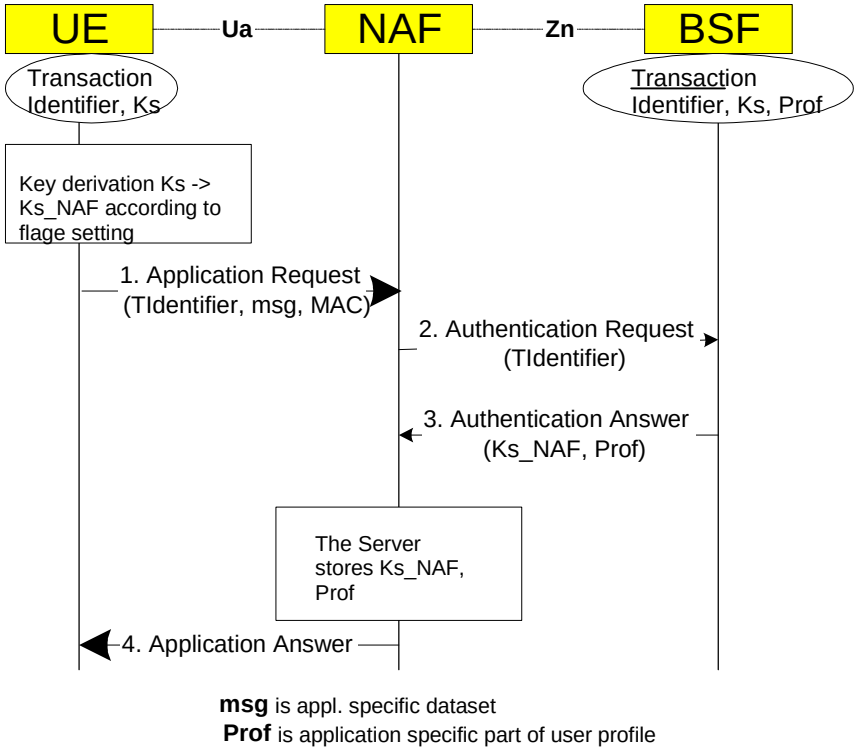


Figure 5: The bootstrapping usage procedure

4.5.4 Procedure related to service discovery

To enable the bootstrapping procedure, a procedure needs to be described on how to discover the location of BSF. It shall be possible to enable the terminal to be configured either manually or automatically via one of the following approaches:

- The address information shall be published via reliable channel. Subscribers shall store all the parameters as part of the initial establishment of IP connectivity. The addresses need to be input only once;
- The address information shall be pushed automatically to the UE over the air interface when the subscription to bootstrapping service is accepted. All the parameters shall be saved in the UE and used the same manner as above. The procedure is specified in [7];
- The location information shall be discovered automatically based on DHCP, after the IP connectivity has been established. The DHCP server shall provide the UE with the domain name of a BSF and the address of a Domain Name Server (DNS) that is capable of resolving the Fully Qualified Domain Name (FQDN) of the BSF. The procedure is specified in TS 23.228 [8].

NOTE: The location of DHCP server may be pushed to UE through the procedure specified in [7].

Annex A (informative): Generic secure message exchange using HTTP Digest Authentication

A.1 Introduction

This annex describes how HTTP Digest Authentication can be used between UE and any NAF where the protocol over Ua interface is based on HTTP messaging.

HTTP Digest Authentication can also be used as a generic authentication and integrity protection method towards any new NAF The Generic Bootstrapping Architecture specified in this document enables the NAF and the UE to mutually authenticate each other and integrity protect any payload being transferred between NAF and UE. As a generic method, it will speed up the specification of new NAFs since the authentication and message integrity protection part of Ua interface are taken care of by HTTP Digest Authentication. It will also ease the implementation of GBA-based authentication in NAFs because there would be one well-defined way to do it.

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 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 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).

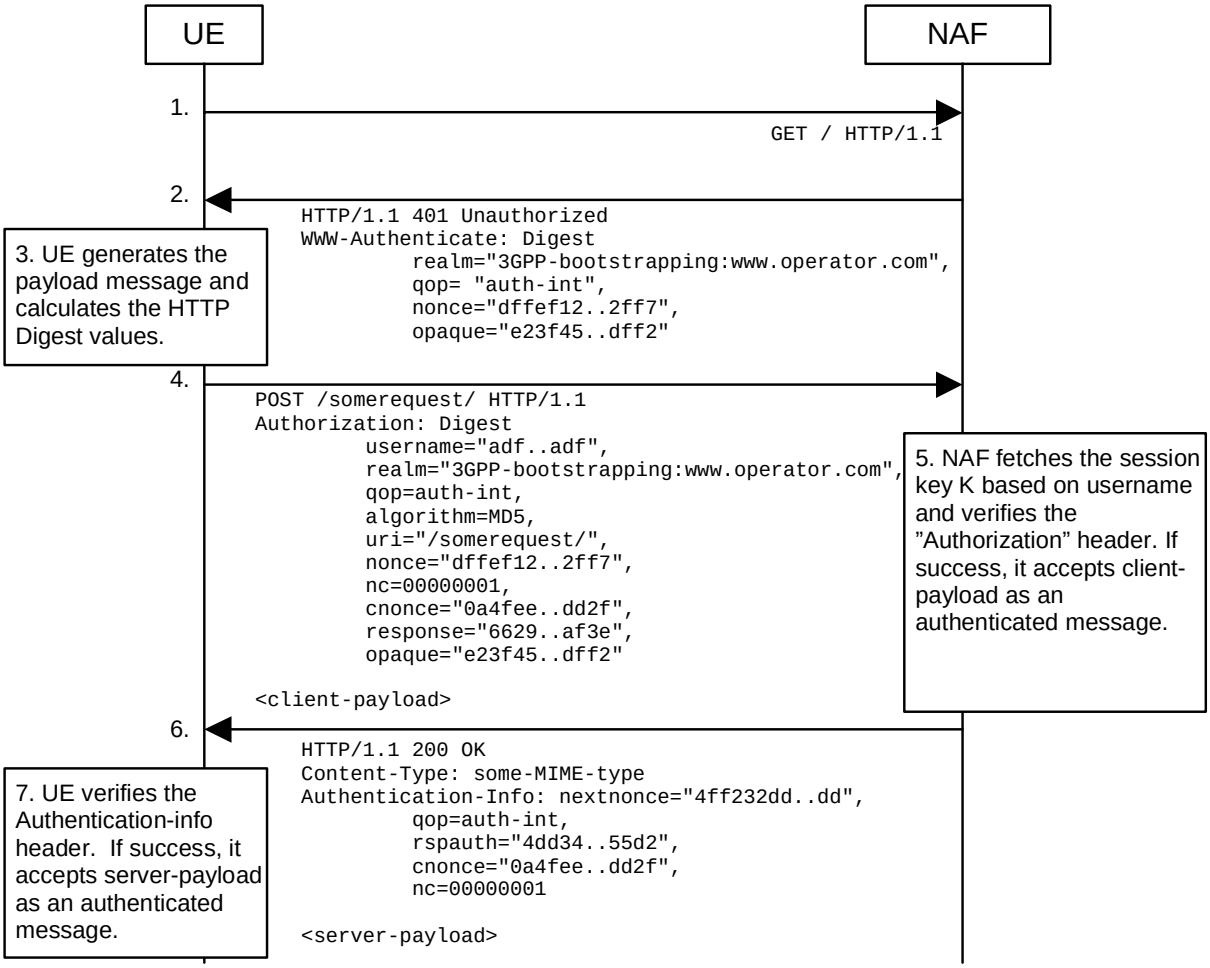


Figure A.1: Generic secure message exchange using HTTP Digest Authentication and bootstrapped security association

Annex B (normative): Specification of the key derivation function KDF

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

Annex C (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2003-10	SA3#30				New draft TS: Generic Bootstrapping Architecture (GBA). Extracted from 33.109 clause 4 and Annex B.		0.1.0
2003-10	SA3#30	S3-030537			New interface names.		0.1.0
2003-10	SA3#30	S3-030538			Requirements for Ub and Zh interfaces added.	0.1.0	0.1.1
2003-10	SA3#30	S3-030545			NAF initiated bootstrapping added	0.1.0	0.1.1
2003-10					Imported Zn interface requirements from SSC TS.	0.1.0	0.1.1
2003-11	SA3#31	S3-030728			Bootstrapping procedure: merging of last two messages	0.1.1	0.2.0
2003-11	SA3#31	S3-030793			Key separation	0.1.1	0.2.0
2003-11	SA3#31	S3-030794			Removal of application specific user profile requirements from GBA	0.1.1	0.2.0
2003-11					Annex A: changed "session key K" to "session key Ks"	0.1.1	0.2.0
2003-12	SP-22	SP-030583	-	-	Presentation to TSG SA#22 for Information	0.2.0	1.0.0
2004-02	SA3#32	S3-040189			Service discovery is added for review of other group	1.0.0	1.1.0
2004-02	SA3#32	S3-040060			Editorial changes throughout the specification	1.1.0	1.2.0
2004-02	SA3#32	S3-040024			Editorial changes throughout the specification	1.1.0	1.2.0
2004-02	SA3#32	S3-040078			Editorial changes on removal of unnecessary texts	1.1.0	1.2.0
2004-02	SA3#32	S3-040157			GBA Transaction Identifier requirements	1.1.0	1.2.0
2004-02	SA3#32	S3-040191			Life time of the bootstrapping information	1.1.0	1.2.0
2004-02	SA3#32	S3-040154			Deletion of parameter n	1.1.0	1.2.0
2004-02	SA3#32	S3-040041			Key handling in the UE in a Generic Bootstrapping Architecture	1.1.0	1.2.0
2004-02	SA3#32	S3-040161			Multiple key derivation in a Generic Bootstrapping Architecture	1.1.0	1.2.0
2004-02					Editorial update for submission to TSG SA for approval	1.2.0	1.2.1
2004-03	SP-23				Editorial update for presentation to TSG SA #23 for approval (MCC)	1.2.0	2.0.0