

Superseded **by a later version of this document**

PacketCable™ 2.0 **IMS Delta Specifications**

Generic Authentication Architecture (GAA); **Generic bootstrapping architecture Specification** **3GPP TS 33.220**

PKT-SP-33.220-I02-061013

ISSUED

Notice

This PacketCable specification is a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. (CableLabs®) for the benefit of the cable industry. Neither CableLabs, nor any other entity participating in the creation of this document, is responsible for any liability of any nature whatsoever resulting from or arising out of use or reliance upon this document by any party. This document is furnished on an AS-IS basis and neither CableLabs, nor other participating entity, provides any representation or warranty, express or implied, regarding its accuracy, completeness, or fitness for a particular purpose.

CableLabs received copyright licenses from ETSI to reproduce, modify, and distribute the 3GPP specifications contained in the PacketCable IMS Delta Specifications. CableLabs will submit these enhancements to the 3GPP for incorporation into the IMS specifications. As this occurs, PacketCable IMS Delta Specifications will be withdrawn and replaced with direct references to 3GPP IMS specifications.

© Copyright 2006 Cable Television Laboratories, Inc.
All rights reserved.

Document Status Sheet

Document Control Number:	PKT-SP-33.220-I02-061013			
Document Title:	Generic Authentication Architecture (GAA); Generic bootstrapping architecture Specification			
Revision History:	I01 – Released 04/06/06 I02 – Released 10/13/06			
Date:	October 13, 2006			
Status:	Work in Progress	Draft	Issued	Closed
Distribution Restrictions:	Author Only	CL/Member	CL/Member/ Vendor	Public

Key to Document Status Codes:

- Work in Progress** An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.
- Draft** A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.
- Issued** A stable document, which has undergone rigorous member and vendor review and is suitable for product design and development, cross-vendor interoperability, and for certification testing.
- Closed** A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks:

DOCSIS®, eDOCSIS™, PacketCable™, CableHome®, CableOffice™, OpenCable™, OCAP™, CableCARD™, M-CMTS™, DCAS™, and CableLabs® are trademarks of Cable Television Laboratories, Inc.

Abstract

This CableLabs-modified 3GPP technical specification includes the cable-specific requirements necessary for implementing 3GPP technical specifications in PacketCable™ 2.0 and the delivery of PacketCable 2.0 services.

Because these are modified 3GPP documents, their document formatting has been retained except as follows. Changes to the original 3GPP requirements are shown in this document by color coding of text. Unchanged text appears normal, while new text appears in blue underline and deleted 3GPP text appears as ~~violet strikethrough hidden text~~. To view the deleted 3GPP text, the reader must have Word configured so the 'view hidden text' is turned on.

The intended audience for this document includes developers of equipment intended to be conformant to PacketCable 2.0 specifications.

NOTE: Special permission has been granted by 3GPP Organizational Partners to reproduce their technical specification, 3GPP TS 33.220, in this document.

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>

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© 2005, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).
All rights reserved.

Contents

Foreword.....	3
1 Scope	3
2 References	3
3 Definitions, abbreviations symbols and conventions	6
3.1 Definitions	6
3.2 Abbreviations	7
3.3 Symbols	7
3.4 Conventions.....	7
4 Generic Bootstrapping Architecture	8
4.1 Reference model.....	8
4.2 Network elements.....	9
4.2.1 Bootstrapping server function (BSF).....	9
4.2.2 Network application function (NAF).....	10
4.2.2a Diameter proxy (D-Proxy).....	10
4.2.3 HSS.....	11
4.2.4 UE.....	11
4.2.5 SLF.....	12
4.3 Bootstrapping architecture and reference points	12
4.3.1 Reference point Ub	12
4.3.2 Reference point Ua	12
4.3.3 Reference point Zh.....	13
4.3.4 Reference point Zn.....	13
4.3.5 Reference point Dz	13
4.4 Requirements and principles for bootstrapping.....	13
4.4.1 Access Independence.....	13
4.4.2 Authentication methods	13
4.4.3 Roaming.....	14
4.4.4 Requirements on reference point Ub.....	14
4.4.5 Requirements on reference point Zh.....	14
4.4.6 Requirements on reference point Zn.....	15
4.4.7 Requirements on Bootstrapping Transaction Identifier	16
4.4.8 Requirements on selection of UICC application and related keys.....	16
4.4.9 Requirements on reference point Ua.....	18
4.4.10 Requirements on reference point Dz.....	18
4.5 Procedures	18
4.5.1 Initiation of bootstrapping.....	18
4.5.2 Bootstrapping procedures	19
4.5.3 Procedures using bootstrapped Security Association	21
4.5.4 Procedure related to service discovery.....	24
5 UICC-based enhancements to Generic Bootstrapping Architecture (GBA_U).....	24
5.1 Architecture and reference points for bootstrapping with UICC-based enhancements.....	24
5.2 Requirements and principles for bootstrapping with UICC-based enhancements	24
5.2.1 Requirements on UE.....	25
5.2.2 Requirements on BSF	25
5.3 Procedures for bootstrapping with UICC-based enhancements	25
5.3.1 Initiation of bootstrapping.....	25
5.3.2 Bootstrapping procedure.....	25

5.3.3	Procedures using bootstrapped Security Association	27
5.3.4	Procedure related to service discovery.....	30
6	HTTP Digest Over TLS enhancements to Generic Bootstrapping Architecture (GBA_H)	30
6.1	Bootstrapping Procedure	30
6.2	Procedures Using Bootstrapped Security Association	32
Annex A: Void	35
Annex B (normative): Specification of the key derivation function KDF		36
B.1	Introduction	36
B.2	Generic key derivation function	36
B.2.1	Input parameter encoding	36
B.3	NAF specific key derivation in GBA _A , GBA _U , and GBA_H	37
Annex C (informative): Void		39
Annex D (informative): Dialog example for user selection of UICC application used in GBA		40
Annex E (normative): TLS profile for securing Zn' reference point		41
Annex F (informative): Handling of TLS certificates		42
F.1	TLS certificate enrollment	42
F.2	TLS Certificate revocation	42
Annex G (normative): GBA_U UICC-ME interface		44
G.1	GBA _U Bootstrapping procedure	44
G.2	GBA _U NAF Derivation procedure	44
Annex H (normative): Ua security protocol identifier		46
H.1	Definition.....	46
H.2	Organization Octet.....	46
H.3	Ua security protocol identifiers for 3GPP specified protocols	46
Appendix I	CableLabs Acknowledgements	48
Appendix II	Change History	49

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP) [and further modified by CableLabs](#).

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 [updated and](#) re-released by [CableLabs](#), ~~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 [and from HTTP Digest over TLS](#). 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 HTTP Digest over TLS function](#), an architecture overview and the detailed procedures [on](#) how to bootstrap the credential.

Clause 4 of this specification describes a mechanism, called GBA_ME, to bootstrap authentication and key agreement, which does not require any changes to the UICC. Clause 5 of this specification describes a mechanism, called GBA_U, to bootstrap authentication and key agreement, which does require changes to the UICC, but provides enhanced security by storing certain derived keys on the UICC. [Clause 6 of this specification describes the HTTP Digest over TLS mechanism](#).

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*.
- PacketCable defines several specifications which are based on 3GPP technical specifications. These PacketCable specifications are commonly referred to as PacketCable Delta specifications. For references within this specification which have a corresponding PacketCable Delta specification, the PacketCable Delta specification must be used. The list of PacketCable Delta specifications is:

PKT-SP-23.008	PKT-SP-29.228
PKT-SP-23.218	PKT-SP-29.229
PKT-SP-23.228	PKT-SP-33.203
PKT-SP-24.229	PKT-SP-33.210
PKT-SP-29.109	PKT-SP-33.220

References which have corresponding delta specifications are highlighted with an * below.

- [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)".
- [9] IETF RFC 3546 (2003): "Transport Layer Security (TLS) Extensions".
- [10] 3GPP TS 31.103: "3rd Generation Partnership Project; Technical Specification Group Terminals; Characteristics of the IP Multimedia Services Identity Module (ISIM) application".
- [11] 3GPP TS 23.003: "3rd Generation Partnership Project; Technical Specification Group Core Network; Numbering, addressing and identification".
- [12] IETF RFC 3548 (2003): "The Base16, Base32, and Base64 Data Encodings".
- [13] *3GPP TS 33.210: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Network domain security; IP network layer security".
- [14] IETF RFC 3588 (2003): "Diameter Base Protocol".

- [15] 3GPP TS 31.101: "3rd Generation Partnership Project; Technical Specification Group Terminals; UICC-terminal interface; Physical and logical characteristics".
- [16] 3GPP TS 33.203: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G security; Access security for IP-based services".
- [17] IETF RFC 3280 (2002): "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".
- [18] IETF RFC 2818 (2000): "HTTP over TLS".
- [19] 3GPP TS 33.310: "3rd Generation Partnership Project; Technical Specification Group Service and System Aspects; Network Domain Security (NDS); Authentication Framework (AF)".
- [20] IETF RFC 2560 (1999): "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP".
- [21] FIPS PUB 180-2 (2002): "Secure Hash Standard".
- [22] IETF RFC 2104 (1997): "HMAC: Keyed-Hashing for Message Authentication".
- [23] ISO/IEC 10118-3:2004: "Information Technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions".
- [24] IETF RFC 3629 (2003): "UTF-8, a transformation format of ISO 10646".
- [25] 3GPP TS 33.222: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".
- [26] 3GPP TS 33.246: "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)".
- [27] [IETF RFC 4279 \(2005\): "Pre-Shared Key Ciphersuites for Transport Layer Security \(TLS\)"](#), ~~IETF Internet Draft: "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", December 2005, URL: <http://www.ietf.org/internet-drafts/draft-ietf-tls-psk-09.txt>~~.
- [28] IETF RFC 2246 (1999): "The TLS Protocol Version 1".
- [29] 3GPP TS 24.109: 3rd Generation Partnership Project; Technical Specification Group Core Network and Terminals; Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details".
- [30] [IETF RFC 2782 \(2000\): "A DNS RR for specifying the location of services \(DNS SRV\)".](#)
- [31] [IETF RFC 1750 \(1994\): "Randomness Recommendations for Security".](#)

3 Definitions, abbreviations symbols and conventions

3.1 Definitions

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

Application: In all places in this document where the term application is used to refer to a service offered by the MNO or a third party to the mobile subscriber, then it always denotes the type of application and not the actual instance of an application installed on an application server.

Bootstrapping Server Function: BSF is hosted in a network element under the control of an MNO. BSF, HSS, and UEs participate in GBA in which a shared secret is established between the network and a UE by running the bootstrapping procedure. The shared secret can be used between NAFs and UEs, for example, for authentication purposes.

Bootstrapping Usage Procedure: A procedure using bootstrapped security association over Ua reference point.

GBA Function: A function on the ME executing the bootstrapping procedure with BSF (i.e., supporting the Ub reference point) and providing Ua applications with security association to run bootstrapping usage procedure. GBA function is called by a Ua application when a Ua application wants to use bootstrapped security association.

[HTTP Digest over TLS-based GBA: This is a GBA that uses HTTP Digest over TLS.](#)

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

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

Network Application Function: NAF is hosted in a network element. GBA may be used between NAFs and UEs for authentication purposes, and for securing the communication path between the UE and the NAF.

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

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

NAF Group: A grouping of NAFs to allow assignment of different USSs to NAFs representing the same application. This grouping is done in each home network separately, i.e., one NAF contacting BSFs in different home networks belongs to different groups in every home network.

NAF_Id: The full DNS name of the NAF, concatenated with the Ua security protocol identifier.

Ua Application: An application on the ME intended to run bootstrapping usage procedure with a NAF.

Ua security protocol identifier: An identifier which is associated with a security protocol over Ua.

User Security Setting: A USS is an application and subscriber specific parameter set that defines two parts, an authentication part, which contains the list of identities of the user needed for the application (e.g., IMPUs, MSISDN, pseudonyms), and an authorisation part, which contains the user permission flags (e.g., access to application allowed, type of certificates which may be issued). Sometimes also called application-specific user security setting. The USS is delivered to the BSF as a part of GUSS from the HSS, and from the BSF to the NAF if requested by the NAF.

3.2 Abbreviations

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

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

3.3 Symbols

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

	Concatenation
⊕	Exclusive or

3.4 Conventions

All data variables in this specification are presented with the most significant substring on the left hand side and the least significant substring on the right hand side. A substring may be a bit, byte or other arbitrary length bitstring. Where a variable is broken down into a number of substrings, the leftmost (most significant) substring is numbered 0, the next most significant is numbered 1, and so on through to the least significant.

4 Generic Bootstrapping Architecture

The 3GPP authentication infrastructure, including the 3GPP Authentication Centre (AuC), the USIM or the ISIM, 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 4.1 shows a simple network model of the entities involved in the bootstrapping approach, and the reference points used between them.

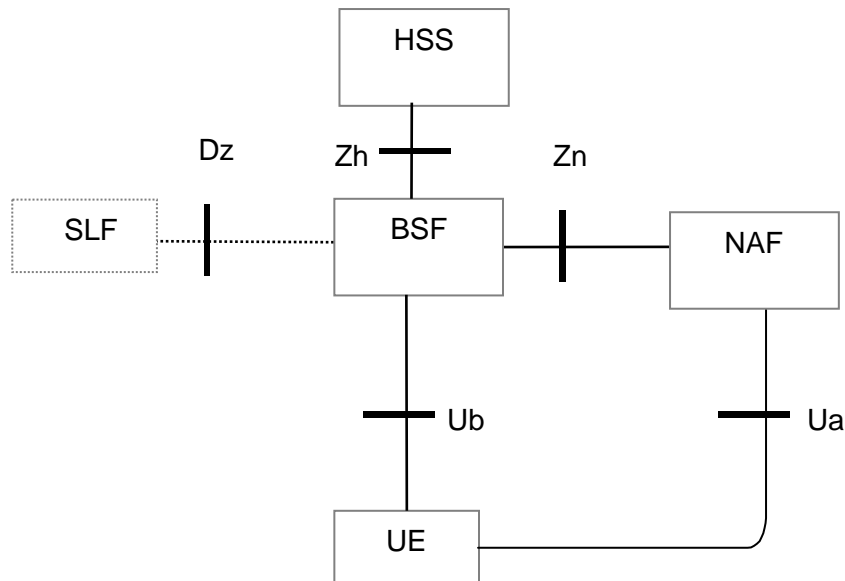
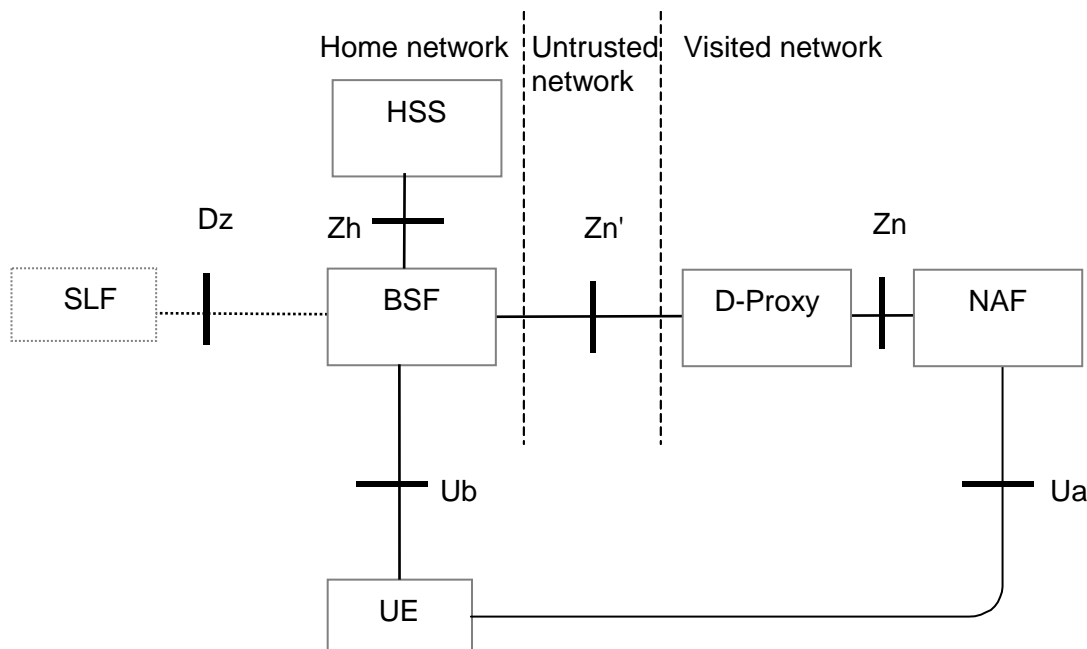


Figure 4.1: Simple network model for bootstrapping

Figure 4.1a shows a simple network model of the entities involved when the network application function is located in the visited network.



NOTE: The Zn' reference point is distinguished from the Zn reference point in that it is used between operators.

Figure 4.1a: Simple network model for bootstrapping in visited network

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 [or the HTTP Digest over TLS mechanisms](#), and agree on session keys that are afterwards applied between UE and a Network Application Function (NAF). The BSF shall restrict the applicability of the key material to a specific NAF by using the key derivation procedure as specified in Annex B. The key derivation procedure may be used with multiple NAFs during the lifetime of the key material. The lifetime of the key material is set according to the local policy of the BSF. The generation of key material is specified in clause 4.5.2.

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

The BSF shall be able to keep a list, which assigns NAFs to NAF Groups. This list is used to select if any and which application-specific USS within GUSS is valid for a certain NAF.

NOTE 1: The operator does the assignment of NAFs to NAF Groups. NAF Group definitions in HSS and all connected BSFs belonging to the same operator's network shall be equal (cf., clause 4.2.3). As these network elements belong to the same operator's network, standardisation of the NAF Group definitions themselves is not necessary in 3GPP.

NOTE 2: The NAF grouping may be e.g., "home" and "visited". It allows the BSF to send USSs for the same application with e.g., different authorization flags to different NAFs, e.g., in home network and visited networks. The NAF e.g., in visited network indicates only the requested application, but it is unaware of the grouping in home network of the subscriber.

4.2.2 Network application function (NAF)

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

General assumptions for the functionality of a NAF are:

- there is no previous security association between the UE and the NAF;
- NAF shall be able to locate and communicate securely with the subscriber's BSF;
- NAF shall be able to acquire a shared key material established between UE and the BSF during the run of the application-specific protocol;
- NAF shall be able to acquire zero or more application-specific USSs from the HSS via the BSF;
- NAF shall be able to set the local validity condition of the shared key material according to the local policy;
- NAF shall be able to check lifetime and local validity condition of the shared key material.

NOTE: Without additional measures, GBA does not guarantee the freshness of the key, $Ks(\text{int/ext})_{\text{NAF}}$ in the sense that it does not guarantee that the key was not used in a previous run of the Ua protocol. The additional measures which may be taken by the UE and the NAF to ensure key freshness in GBA are:

- 1) enforce a new run of the Ub protocol (thus generating a new Ks) before deriving a new Ks_{NAF} .
- 2) store previously used keys $Ks(\text{int/ext})_{\text{NAF}}$, or the corresponding key identifiers B-TID, until the end of their lifetime.

A UE and a NAF that support a Ua protocol that does not provide replay protection over unconnected runs of the protocol, will need to take corresponding action to avoid replay attacks if desired.

4.2.2a Diameter proxy (D-Proxy)

In the case where UE has contacted an NAF that is operated in another network than home network, this visited NAF shall use a diameter proxy (D-Proxy) of the NAFs network to communicate with subscriber's BSF (i.e., home BSF).

NOTE: D-Proxy functionality may be implemented as a separate network element, or be part of any NE in the visited network that implements Diameter proxy functionality (examples of such NE's are the BSF of the network that the visited NAF belongs to, or an AAA-server).

General requirements for the functionality of D-Proxy are:

- D-Proxy shall be able to function as a proxy between the visited NAF, and the subscriber's home BSF;
- D-Proxy shall be able to locate subscriber's home BSF and communicate with it over secure channel;
- D-Proxy shall be able to validate that the visited NAF is authorized to participate in GBA and shall be able to assert to subscriber's home BSF the visited NAFs DNS name. The D-Proxy shall also be able to assert to the BSF that the visited NAF is authorized to request the GBA specific user profiles contained in the NAF request;
- the physical security level of the D-proxy shall not be lower than the highest level of the NAFs which it interfaces with.

4.2.3 HSS

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

The requirements on the HSS are:- HSS shall provide the only persistent storage for GUSSs;

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

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

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

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

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

- HSS shall be able to assign application-specific USSs to a NAF Group. This shall be defined in such a way that different USSs for the same application, but for different groups of NAFs, are possible. The restrictions on the number of USSs per GUSS are dependent on the usage of NAF Groups by the operator:
 - if no NAF Groups are defined for this application then at most one USS per application is stored in GUSS;
 - if NAF Groups are defined for this application then at most one USS per application and NAF Group is stored in GUSS.
- NAF Group definitions in the HSS and all connected BSFs belonging to the same operator's network shall be equal.

4.2.4 UE

The required functionalities from the UE [that supports a UICC](#) are:

- the support of HTTP Digest AKA protocol;
- the capability to use both a USIM and an ISIM in bootstrapping;
- the capability to select either a USIM or an ISIM to be used in bootstrapping, when both of them are present;

- the capability for a Ua application on the ME to indicate to the GBA Function on the ME the type or the name of UICC application to use in bootstrapping (see clause 4.4.8);
- 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]).

The required functionalities from the UE that does not support a UICC are:

- the support of HTTP Digest over TLS;
- support of NAF-specific application protocol defined in TS 33.222 [25]).

A UE that supports a UICC may support the HTTP Digest over TLS functionality.

A GBA-aware ME with a UICC shall support both GBA_U, as specified in clause 5.2.1 and GBA_ME procedures, as specified in clause 4.5.

4.2.5 SLF

The SLF:

- is queried by the BSF in conjunction with the Zh interface operation to get the name of the HSS containing the required subscriber specific data.
- is accessed via the Dz interface by the BSF.

The SLF is not required in a single HSS environment. Use of SLF is not required when BSF are configured/managed to use pre-defined HSS.

4.3 Bootstrapping architecture and reference points

4.3.1 Reference point Ub

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 or by using HTTP Digest over TLS mechanism.

The HTTP Digest AKA protocol, which is specified in RFC 3310 [4], is used on the reference point Ub. 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] and to the ISIM is as specified in TS 31.103 [10].

The HTTP Digest protocol, which is specified in RFC 2617 [3], in conjunction with TLS is also used on the reference point Ub.

4.3.2 Reference point Ua

The reference point Ua 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 or HTTP Digest over TLS over reference point Ub. 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.

4.3.3 Reference point Zh

The reference point Zh used between the BSF and the HSS allows the BSF to fetch the required authentication information and all GBA user security settings from the HSS. The interface to the 3G Authentication Centre is HSS-internal, and it need not be standardised as part of this architecture.

4.3.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous HTTP Digest AKA protocol [or HTTP Digest over TLS](#) run over the reference point Ub from the UE to the BSF. It is also used to fetch application-specific user security settings from the BSF, if requested by the NAF.

4.3.5 Reference point Dz

The reference point Dz used between the BSF and the SLF allows the BSF to get the name of the HSS containing the required subscriber specific data.

4.4 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 and in the visited network;
- the architecture shall not preclude the support of network application function 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 is using the GBA, can be used by an attacker to mount successful attacks to the other NAFs using the GBA;
- an attacker shall not be able to exploit a security breach in one security protocol over Ua in order to mount a successful attack against a different security protocol over Ua.

4.4.1 Access Independence

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

4.4.2 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 [or HTTP Digest over TLS](#).

4.4.3 Roaming

The requirements on roaming are:

- The roaming subscriber shall be able to utilize the bootstrapping function in the home network. The subscriber shall be able to utilize network application function that is in a visited network.
- The home network shall be able to control whether its subscriber is authorized to use the service in the visited network.

4.4.4 Requirements on reference point Ub

The requirements for reference point Ub 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 and the UE shall be able to authenticate each other based on HTTP Digest over TLS;](#)
- the BSF shall be able to send a bootstrapping transaction identifier to the UE;
- the UE and the BSF shall establish shared keys;
- the BSF shall be able to indicate to the UE the lifetime of the key material. The key lifetime sent by the BSF over Ub shall indicate the expiry time of the key.

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

4.4.5 Requirements on reference point Zh

The requirements for reference point Zh are:

- mutual authentication, confidentiality and integrity shall be provided;

NOTE 1: 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 one 3GPP AKA vector at a time to the BSF;
[the HSS shall be able to send HTTP Digest credentials to the BSF;](#)
- the HSS shall be able to send the complete set of subscriber's GBA user security settings needed for security purposes to the BSF;

NOTE 2: If subscriber's GUSS is updated in HSS, this is not propagated to the BSF. The GUSS in the BSF is updated when the BSF next time fetches the authentication vectors and GUSS from the HSS over Zh reference point as part of the bootstrapping procedure.

- no state information concerning bootstrapping shall be required in the HSS;
- all procedures over reference point Zh shall be initiated by the BSF;
- the number of different interfaces to HSS should be minimized.

4.4.6 Requirements on reference point Zn

The requirements for reference point Zn are:

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

NOTE 1: Annex E specifies the TLS profile that is used for securing the Zn' reference point.

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

NOTE 2: If some application needs only a subset of an application-specific USS, e.g., only one IMPU, the NAF selects this subset from the complete set of USS sent from BSF.

- The BSF shall be able to be configured on a per NAF or per application basis if private subscriber identity and which application-specific USSs may be sent to a NAF;

NOTE 3: If the BSF does not send the IMPI or any other user identities in the USS to the NAF, the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user pseudonym.

- If a NAF requests USSs from the BSF and they are not present in subscriber's GUSS, it shall not cause an error, provided the conditions of the local policy of the BSF are fulfilled. The BSF shall then send only the requested and found USSs to the NAF;
- It shall be possible to configure a local policy as follows: BSF may require one or more application-specific USS to be present in a particular subscriber's GUSS for a particular requesting NAF, and to reject the request from the NAF in case the conditions are not fulfilled. In order to satisfy this local policy, it is not required that the NAF requests the USSs over the Zn reference point, which the BSF requires to be present in the GUSS, rather it is sufficient that the BSF checks the presence of the USSs locally. It shall also be possible to configure the BSF in such a way that no USS is required for the requesting NAF;
- The BSF shall be able to indicate to the NAF the bootstrapping time and the lifetime of the key material. The key lifetime sent by the BSF over Zn shall indicate the expiry time of the key, and shall be identical to the key lifetime sent by the BSF to the UE over Ub.

NOTE 4: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

NOTE 5: If one or more of the USSs that have been delivered to the NAF has been updated in subscriber's GUSS in the HSS, this change is propagated to the NAF the next time it fetches the USS from the BSF over Zn reference point (provided that the BSF has updated subscriber's GUSS from the HSS over Zh reference point).

4.4.7 Requirements on Bootstrapping Transaction Identifier

Bootstrapping transaction identifier (B-TID) shall be used to bind the subscriber identity to the keying material in reference points Ua, Ub and Zn.

Requirements for B-TID are:

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

NOTE 1: NAF can remove the security association based on deletion conditions after the key has become invalid.

NOTE 2: Care has to be taken that the parallel use of GBA and non-GBA authentication between UE and NAF does not lead to conflicts, e.g., in the name space. This potential conflict cannot be resolved in a generic way as it is dependent on specific protocol and authentication mechanism used between UE and application server. It is therefore out of scope of this specification. For the example of HTTP Digest authentication used between UE and NAF, parallel use is possible as the following applies: <username,password>-pairs must be unique to one realm only. As the NAF controls the realm names, it has to ensure that only the GBA based realm is named with the reserved 3GPP realm name. In the special case that the NAF wants to allow non GBA based authentication in the GBA realm also, it has to ensure that no usernames in the format of a B-TID are used outside GBA based authentication.

4.4.8 Requirements on selection of UICC application and related keys

[The requirements in this section apply when a UICC is present in the UE.](#)

When several applications are present on the UICC, which are capable of running AKA, then the ME shall choose one of these UICC applications for performing the GBA procedures specified in this document in the following order of preference:

1. The UE determines which UICC application is to be involved:
 - a. the application on the ME that needs Ks_NAF (Ua application) may indicate to the GBA support function (GBA function) the type or the name of the UICC application: no preference, USIM, ISIM, or the "Label" (see definition in TS 31.101 [15]) of the UICC application.

If the application on the ME indicated a "Label" of the UICC application, step b below shall be executed.

If the application on the ME indicated that the UICC application type should be:

- the USIM on the UICC; step b below is skipped and in steps c and d only USIM applications are considered.
- the ISIM on the UICC; step b below is skipped and in steps c and d only ISIM applications are considered.

if the application on the ME did not indicate a preference, step b below is skipped and the selection process is executed as described below, starting with step c;

- b. if a "Label" was indicated in step a, the GBA function shall select (see definition in TS 31.102 [1]) the UICC application with the "Label" indicated; if selection of this UICC application does not succeed the selection procedure fails;
- c. if no "Label" was indicated in step a, the GBA function shall choose among the active UICC applications; if there is more than one active UICC applications, the GBA function may show a UICC application choosing dialogue to the end user (the list contains the "Labels" from the application list of the UICC), from which the end user chooses the UICC application to be selected; if no dialogue is shown the GBA function shall select the "last selected" active UICC application; in case the Ua application indicated "no preference" and both a "last selected" USIM and a "last selected" ISIM are active, then the "last selected" USIM is selected.
- d. if there are no UICC applications active:
 - if there is only one UICC application, the GBA function selects it, if possible;
 - if there is more than one UICC application, the GBA function may show a UICC application choosing dialogue to the end user (the list contains the "Labels" from the application list of the UICC), from which the end user chooses the UICC application to be selected; if no dialogue is shown the GBA function shall select the "last selected" UICC application, if possible.
- e. if the type indicated in step a and used in step d was ISIM, but there was no ISIM to select, then step d is repeated with type USIM; otherwise the selection process fails.

NOTE 1: Step e is required for the case that an ISIM as defined in TS 33.203 [16] may be realised using a USIM application on the UICC.

2. If there already is a key Ks derived from the chosen UICC application, the UE takes this key to derive Ks_NAF.
3. If there is no such key Ks, the UE first runs the Ub protocol involving the selected UICC application and then goes to step 2.

If a USIM is chosen, the IMPI obtained from the IMSI stored on the USIM as specified in TS 23.003 [11] clause 13.3, is used in the protocol run over Ub.

NOTE 2: Strictly speaking, an IMPI, and the derivation of an IMPI from an IMSI as in TS 23.003 [11], clause 13 are only defined in the context of the IMS. For the purposes of this specification, however, an identifier obtained from an IMSI as specified in TS 23.003 [11], clause 13.3 is also called an IMPI, even if the user has no IMS subscription.

If an ISIM is selected, the IMPI stored on the ISIM is used in the protocol run over Ub.

Whenever a UICC application is successfully selected or terminated, the rules in this clause for choosing the UICC application are re-applied and, consequently, the UICC application chosen for GBA may change.

Whenever a UICC application is terminated the shared key Ks established from it in the protocol over the Ub reference point (according to clauses 4.5.2 and 5.3.2) shall be deleted.

NOTE 3: At any one time, there is at most one UICC application chosen for performing the GBA procedures.

NOTE 4: The Ua applications can continue using the NAF specific keys (Ks_(ext/int)_NAF) derived also after the shared key Ks itself has been deleted until the Ks lifetime expires.

4.4.9 Requirements on reference point Ua

The generic requirements for reference point Ua are:

- the UE and the NAF shall be able to secure the reference point Ua using the GBA-based shared secret;

NOTE 1: The exact method of securing the reference point Ua depends on the application protocol used over reference point Ua.

- any security protocol over Ua shall be associated with a Ua security protocol identifier. This identifier shall be specified in Annex H of this specification.
- the NAF shall be able to indicate to the UE that GBA-based shared secret should be used;
- the NAF shall be able to indicate to the UE that the current shared secret has expired and the UE should use newer shared secret with the NAF.
- The default lifetime of the NAF specific key material $Ks_{(ext/int)}_{NAF}$ shall be equal to the lifetime of Ks when not specified within the Ua-application specification. The lifetime of the $Ks_{(ext/int)}_{NAF}$ shall not exceed the lifetime of corresponding Ks .
- The UE and NAF may adapt the key material $Ks_{(ext/int)}_{NAF}$ to the specific needs of the reference point Ua. This adaptation is outside the scope of this specification. The default lifetime of the adapted key material shall be equal to the lifetime of $Ks_{(ext/int)}_{NAF}$ when not specified within the Ua-application specification. The lifetime of the adapted key material shall not exceed the lifetime of corresponding $Ks_{(ext/int)}_{NAF}$.

4.4.10 Requirements on reference point Dz

This interface between BSF and SLF is used to retrieve the address of the HSS which holds the subscription for a given user. This interface is not required in a single HSS environment.

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

Before communication between the UE and the NAF can start, the UE and the NAF first have to agree whether to use the GBA. When a UE wants to interact with a NAF, but it does not know if the NAF requires the use of shared keys obtained by means of the GBA, the UE shall contact the NAF for further instructions (see figure 4.2).

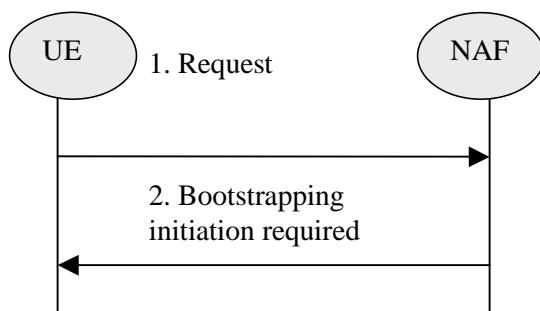


Figure 4.2: Initiation of bootstrapping

1. UE starts communication over reference point Ua with the NAF without any GBA-related parameters.
2. If the NAF requires the use of shared keys obtained by means of the ~~GBA~~, but GBA, the request from UE does not include GBA-related parameters, the NAF replies with a bootstrapping initiation message. The form of this indication may depend on the particular reference point Ua and is specified in the relevant stage 3-specifications.

4.5.2 Bootstrapping procedures

[The requirements in this section apply when a UE is performing AKA.](#)

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 4.3). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping negotiation indication from the NAF, or when the lifetime of the key in UE has expired (cf. subclause 4.5.3).

NOTE 1: 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.

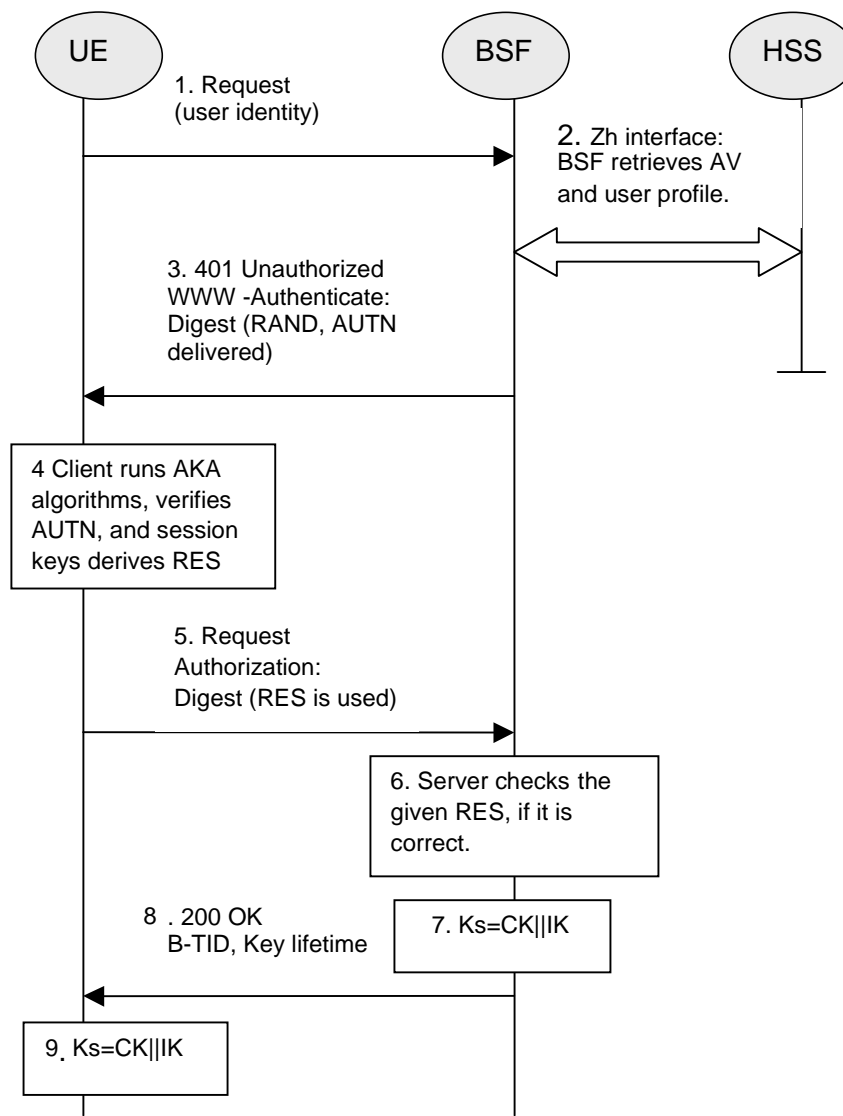


Figure 4.3: The bootstrapping procedure

1. The UE sends an HTTP request towards the BSF.
2. BSF retrieves the complete set of GBA user security settings and one Authentication Vector (AV, AV = RAND||AUTN||XRES||CK||IK) over the reference point Zh from the HSS.

NOTE 2: In a multiple HSS environment, the BSF may have to obtain the address of the HSS where the subscription of the user is stored by querying the SLF, prior to step 2.

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 B-TID value shall be also generated in format of NAI by taking the base64 encoded [12] RAND value from step 3, and the BSF server name, i.e., `base64encode(RAND)@BSF_servers_domain_name`.
8. The BSF shall send a 200 OK message, including a B-TID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key Ks. 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 during the procedures as specified in clause 4.5.3. Ks_NAF shall be used for securing the reference point Ua.

Ks_NAF is computed as $Ks_NAF = KDF(Ks, "gba-me", RAND, IMPI, NAF_Id)$, where KDF is the key derivation function as specified in Annex B, and the key derivation parameters consist of the user's IMPI, the NAF_Id and RAND. The NAF_Id consists of the full DNS name of the NAF, concatenated with the Ua security protocol identifier as specified in Annex H. KDF shall be implemented in the ME.

NOTE 3: To allow consistent key derivation based on NAF name in UE and BSF, at least one of the three following prerequisites shall be fulfilled:

- (1) The NAF is known in DNS under one domain name (FQDN) only, i.e., no two different domain names point to the IP address of the NAF. This has to be achieved by administrative means. This prerequisite is not specific to 3GPP, as it is necessary also under other circumstances, e.g., for TLS V1.0 without use of wildcard or multiple-name certificates.
- (2) Each DNS entry of the NAF points to a different IP address. The NAF responds to all these IP addresses. Each IP address is tied to the corresponding FQDN by NAF configuration. The NAF can see from the IP address, which FQDN to use for key derivation.
- (3) Ua uses a protocol which transfers the host name (FQDN of NAF as used by UE) to NAF (e.g., HTTP/1.1 with mandatory Host request header field). This requires the NAF to check the validity of the host name, to use this name in all communication with UE where appropriate, and to transfer this name to BSF to allow for correct derivation of Ks_NAF.
In case of a TLS tunnel this requires either multiple-identities certificates or the deployment of RFC 3546 [9] or other protocol means with similar purpose.

The UE and the BSF shall store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated.

4.5.3 Procedures using bootstrapped Security Association

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

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

1. UE starts communication over reference point Ua with the NAF:
 - in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e., if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available), the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:

- if a key K_s for the selected UICC application is available in the UE, the UE derives the key K_{s_NAF} from K_s , as specified in clause 4.5.2;
- if no key K_s for the selected UICC application is available in the UE, the UE first agrees on a new key K_s with the BSF over the reference point U_b , and then proceeds to derive K_{s_NAF} ;

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

- if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g., because the key's lifetime has expired or will expire soon, or the key can not meet the NAF local validity condition, it shall send a suitable bootstrapping renegotiation request to the UE, see figure 4.5. If the key's lifetime has expired the protocol used over reference point U_a shall be terminated. The form of this indication depends on the particular protocol used over reference point U_a . If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point U_b , as specified in clause 4.5.2, in order to obtain a new key K_s .

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

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

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

NOTE 4: The UE may adapt the key material K_{s_NAF} to the specific needs of the reference point U_a . This adaptation is outside the scope of this specification.

- key management for GBA related keys in the ME (i.e., K_s and K_{s_NAF} keys):
 - all GBA related keys shall be deleted from the ME when a different UICC is inserted. Therefore the ME needs to store in non-volatile memory the last inserted UICC-identity to be able to compare that with the used UICC-identity at UICC insertion and power on;
 - the Key K_s shall be deleted from the ME when the ME is powered down;
 - all other GBA related keys may be deleted from the ME when the ME is powered down. If the ME does not delete the GBA keys at power down then the GBA keys need to be stored in non-volatile memory.
- when a new K_s is agreed over the reference point U_b and a key K_{s_NAF} , derived from one NAF_Id , is updated, the other keys K_{s_NAF} , derived from different values NAF_Id , stored on the UE shall not be affected;

NOTE 5: According to the procedures defined in clauses 4.5.2 and 4.5.3, in the UE there is at most one K_{s_NAF} key stored per NAF_Id .

2. NAF starts communication over reference point Z_n with BSF

- The NAF requests key material corresponding to the B-TID supplied by the UE to the NAF over reference point U_a . If the NAF has several FQDNs, which may be used in conjunction with this specification, then the NAF shall transfer in the request over Z_n the same FQDN, which was used over U_a (see NOTE 2 on key derivation in this clause);
- The NAF may also request one or more application-specific USSs for the applications, which the request received over U_a from UE may access;

- With the key material request, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able verify that NAF is authorized to use that hostname;
3. The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause 4.5.2, and supplies to NAF the requested key Ks_NAF, as well as the bootstrapping time and the lifetime of that key, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.
- NOTE 6: The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.
- NOTE 7: The NAF will adapt the key material Ks_NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.
- The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause 4.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.
 - The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy;
4. NAF continues with the protocol used over the reference point Ua with the UE.

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

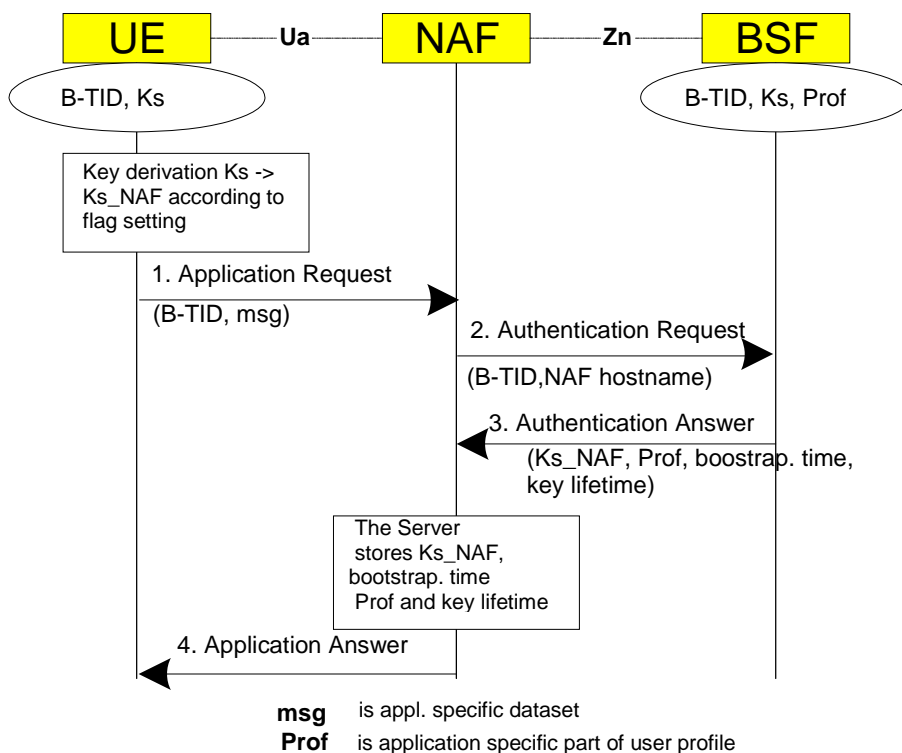


Figure 4.4: The bootstrapping usage procedure

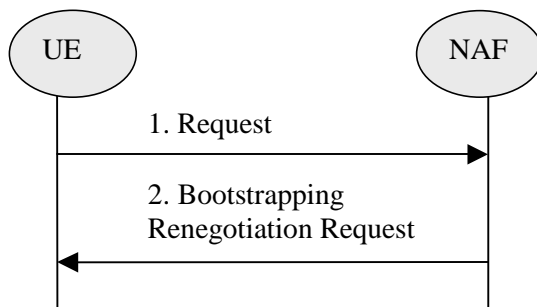


Figure 4.5: Bootstrapping renegotiation request

4.5.4 Procedure related to service discovery

[BSF discovery may occur in one of several ways:](#)

~~The~~ When using the HTTP Digest AKA method, the UE ~~may shall~~ discover the address of the BSF the from the identity information related to the UICC application that is used during bootstrapping procedure, i.e., IMSI for USIM, or IMPI for ISIM. The address of the BSF is derived as specified in 3GPP TS 23.003 [11].

- [A UE may discover the BSF through the use of DNS SRV as defined in RFC 2782 \[30\]. The UE shall have knowledge of the DNS SRV service name and the protocol for the BSF it is trying to locate. The procedures of RFC 2782 shall be followed to determine the address of the BSF.](#)
- [A UE may be configured with the address of the BSF.](#)

5 UICC-based enhancements to Generic Bootstrapping Architecture (GBA_U)

It is assumed that the UICC, BSF, and HSS involved in the procedures specified in this clause are capable of handling the GBA_U specific enhancements. The procedures specified in this clause also apply if NAF is not GBA_U aware.

5.1 Architecture and reference points for bootstrapping with UICC-based enhancements

The text from clause 4.4 of this specification applies also here, with the addition that the interface between the ME and the UICC, as specified in TS 31.102 [1] and TS 31.103 [10], needs to be enhanced with GBA_U specific commands. The requirements on these commands can be found in clause 5.2.1, details on the procedures are in clause 5.3.

5.2 Requirements and principles for bootstrapping with UICC-based enhancements

The requirements and principles from clause 4.3 also apply here with the following addition:

5.2.1 Requirements on UE

The 3G AKA keys CK and IK resulting from a run of the protocol over the Ub reference point shall not leave the UICC.

The UICC shall be able to distinguish between authentication requests for GBA_U, and authentication requests for other 3G authentication domains.

Upon an authentication request from the ME, which the UICC recognises as related to GBA_U, the UICC shall derive the bootstrapping key.

Upon request from the ME, the UICC shall be able to derive further NAF-specific keys from the derived key stored on the UICC.

All GBA-aware MEs shall support procedures for the two previous requests.

5.2.2 Requirements on BSF

BSF shall support both GBA_U and GBA_ME bootstrapping procedures. The decision on running one or the other shall be based on subscription information (i.e., UICC capabilities).

The BSF shall be able to acquire the UICC capabilities related to GBA as part of the GBA user security settings received from the HSS.

5.3 Procedures for bootstrapping with UICC-based enhancements

5.3.1 Initiation of bootstrapping

The text from clause 4.5.1 of this document applies also here.

5.3.2 Bootstrapping procedure

The procedure specified in this clause differs from the procedure specified clause 4.5.2 in the local handling of keys and Authentication Vectors in the UE and the BSF. The messages exchanged over the Ub reference point are identical for both 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 5.1). Otherwise, the UE shall perform a bootstrapping authentication only when it has received bootstrapping initiation required message or a bootstrapping renegotiation indication from the NAF, or when the lifetime of the key in UE has expired (see clause 5.3.3).

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 5.1 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.

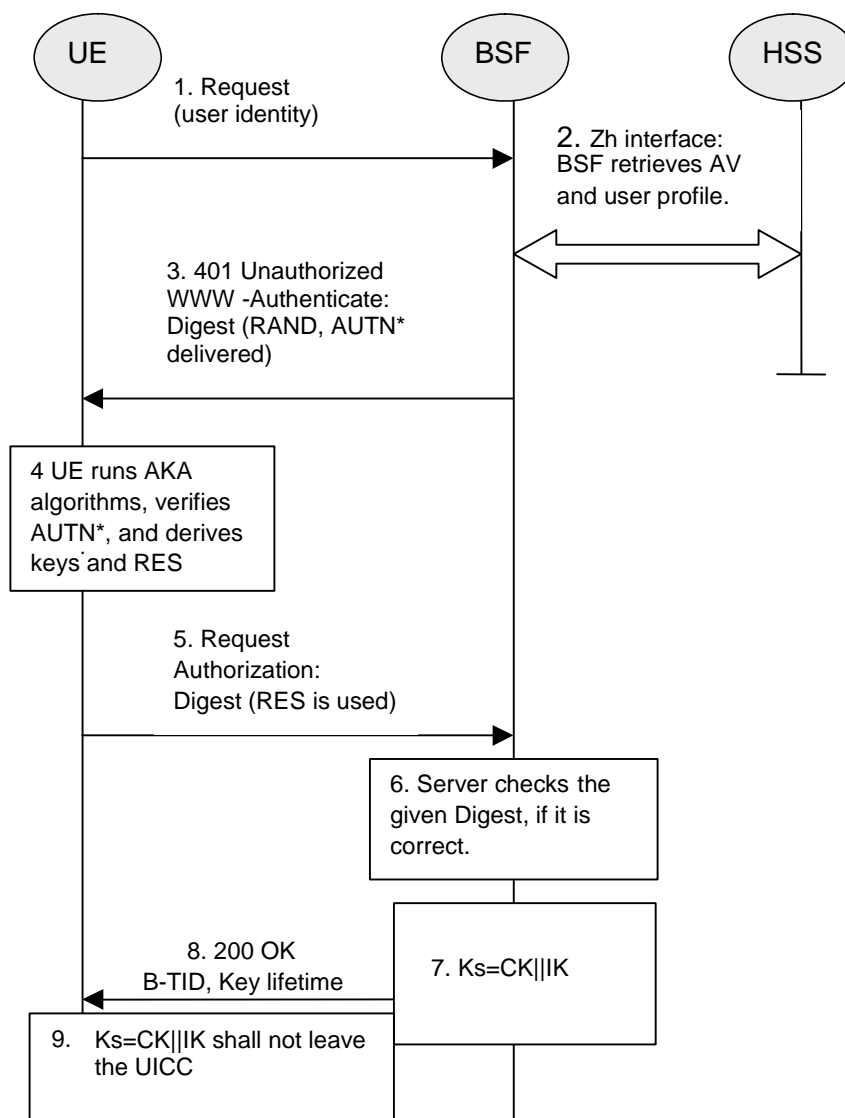


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

1. The ME sends an HTTP request towards the BSF.
2. The BSF retrieves the complete set of GBA user security settings and one Authentication Vector (AV, AV = RAND||AUTN||XRES||CK||IK) over the Zh reference point from the HSS. The BSF can then decide to perform GBA_U, based on the user security settings (USSs). In this case, the BSF proceeds in the following way:

- BSF computes $MAC^* = MAC \oplus \text{Trunc}(\text{SHA-1}(\text{IK}))$

NOTE 1: Trunc denotes that from the 160 bit output of SHA-1 [21], the 64 bits numbered as [0] to [63] are used within the * operation to MAC.

The BSF stores the XRES after flipping the least significant bit.

NOTE 2: In a multiple HSS environment, the BSF may have to obtain the address of the HSS where the subscription of the user is stored by querying the SLF, prior to step 2.

3. Then BSF forwards the RAND and AUTN* (where $AUTN^* = SQN \oplus AK \parallel AMF \parallel MAC^*$) to the UE in the 401 message (without the CK, IK and XRES). This is to demand the UE to authenticate itself.
4. The ME sends RAND and AUTN* to the UICC. The ~~UICC calculates~~ UICC calculates IK and MAC (by performing $MAC = MAC^* \oplus \text{Trunc}(\text{SHA-1}(\text{IK}))$). Then the UICC checks AUTN (i.e., $SQN \oplus AK \parallel AMF \parallel MAC$) to verify that the challenge is from an authorised network; the UICC also calculates CK and RES. This will result in session keys CK and IK in both BSF and UICC. The UICC then transfers RES (after flipping the least significant bit) to the ME and stores Ks, which is the concatenation of CK and IK, on the UICC.
5. The ME 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 the key Ks by concatenating CK and IK. The B-TID value shall be also generated in format of NAI by taking the base64 encoded [12] RAND value from step 3, and the BSF server name, i.e., $\text{base64encode}(\text{RAND})@BSF_servers_domain_name$.
8. The BSF shall send a 200 OK message, including the B-TID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the BSF shall supply the lifetime of the key Ks.
9. Both the UICC and the BSF shall use the Ks to derive NAF-specific keys Ks_ext_NAF and Ks_int_NAF during the procedures as specified in clause 5.3.3, if applicable. Ks_ext_NAF and Ks_int_NAF are used for securing the Ua reference point.

Ks_ext_NAF is computed in the UICC as $Ks_ext_NAF = \text{KDF}(Ks, "gba-me", RAND, IMPI, NAF_Id)$, and Ks_int_NAF is computed in the UICC as $Ks_int_NAF = \text{KDF}(Ks, "gba-u", RAND, IMPI, NAF_Id)$, where KDF is the key derivation function as specified in Annex B, and the key derivation parameters include the user's IMPI, the NAF_Id and RAND. The NAF_Id consists of the full DNS name of the NAF, concatenated with the Ua security protocol identifier as specified in Annex H. The key derivation parameters used for Ks_ext_NAF derivation must be different from those used for Ks_int_NAF derivation. This is done by adding a static string "gba-me" in Ks_ext_NAF and "gba-u" in Ks_int_NAF as an input parameter to the key derivation function.

NOTE 3: The NOTE 3 of clause 4.5.2 also applies here.

The UICC and the BSF store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated.

5.3.3 Procedures using bootstrapped Security Association

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- key management for GBA related keys in the ME (i.e., Ks_ext_NAF keys):
 - all GBA related keys shall be deleted from the ME when a different UICC is inserted. Therefore the ME needs to store in non-volatile memory the last inserted UICC-identity to be able to compare that with the used UICC-identity at UICC insertion and power on.
 - all GBA related keys may be deleted from the ME when the ME is powered down. If the ME does not delete the GBA keys at power down then the GBA keys need to be stored in non-volatile memory.
 - all GBA related keys in the UICC do not need to be deleted when the ME is powered down.

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

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

NOTE 8: According to the procedures defined in clauses 5.3.2 and 5.3.3, in the UE there is at most one Ks_int_NAF/Ks_ext_NAF key pair stored per NAF_Id.

NOTE 9: This rule ensures that the keys Ks_ext_NAF and Ks_int_NAF are always in synch at the UE and the NAF.

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

- The NAF requests from the BSF the keys corresponding to the B-TID, which was supplied by the UE to the NAF over the Ua reference point. If the NAF is GBA_U aware it indicates this by including a corresponding flag in the request. If the NAF has several FQDNs, which may be used in conjunction with this specification, then the NAF shall transfer in the request over Zn the same FQDN, which was used over Ua (see note above on key derivation in this clause).
- The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;
- With the keys request over the Zn reference point, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able to verify that NAF is authorized to use that hostname.

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

NOTE 10: The NAF can further set the local validity condition of the Ks_NAF according to the local policy, for example a limitation of reuse times of a Ks_NAF.

NOTE 11: The NAF will adapt the keys Ks_ext_NAF and Ks_int_NAF to the specific needs of the Ua reference point in the same way as the UE did. This adaptation is outside the scope of this specification.

- The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause 4.4.6). If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.
- The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy.

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

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

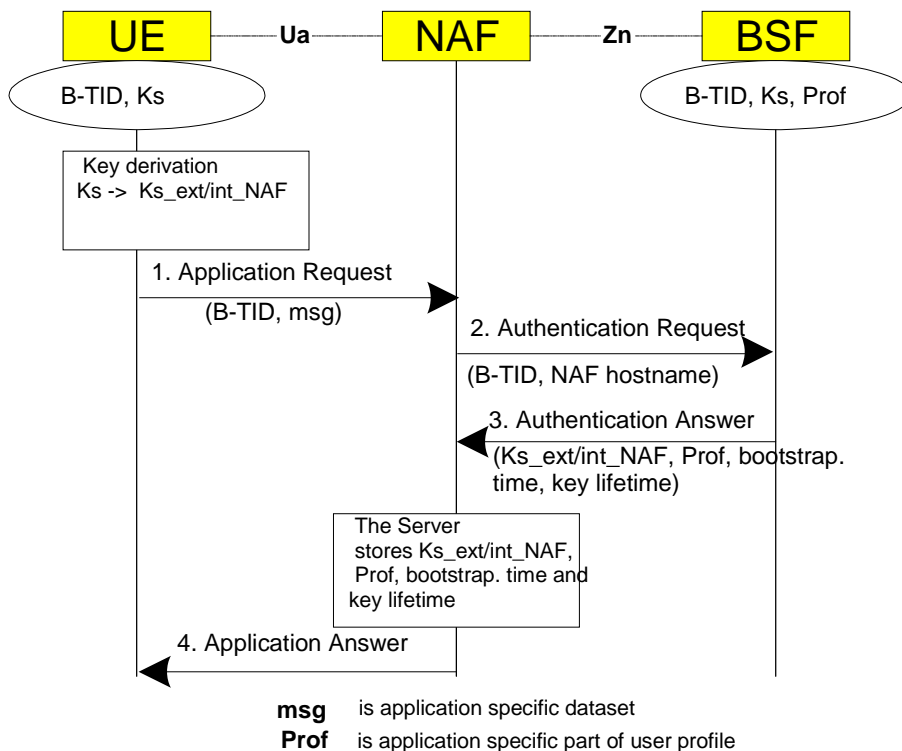


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

5.3.4 Procedure related to service discovery

The text from clause 4.5.4 of this document applies also here.

6 HTTP Digest Over TLS enhancements to Generic Bootstrapping Architecture (GBA_H)

The following sections describe procedures for UEs performing GBA_H, which is based on HTTP Digest over TLS.

6.1 Bootstrapping Procedure

The bootstrapping procedure starts by establishing a TLS tunnel between the client and the BSF. After establishing the TLS tunnel, the Ub interface shall use the HTTP Digest mechanism to establish the credentials (i.e., derive a session key(s)) between the UE and the BSF.

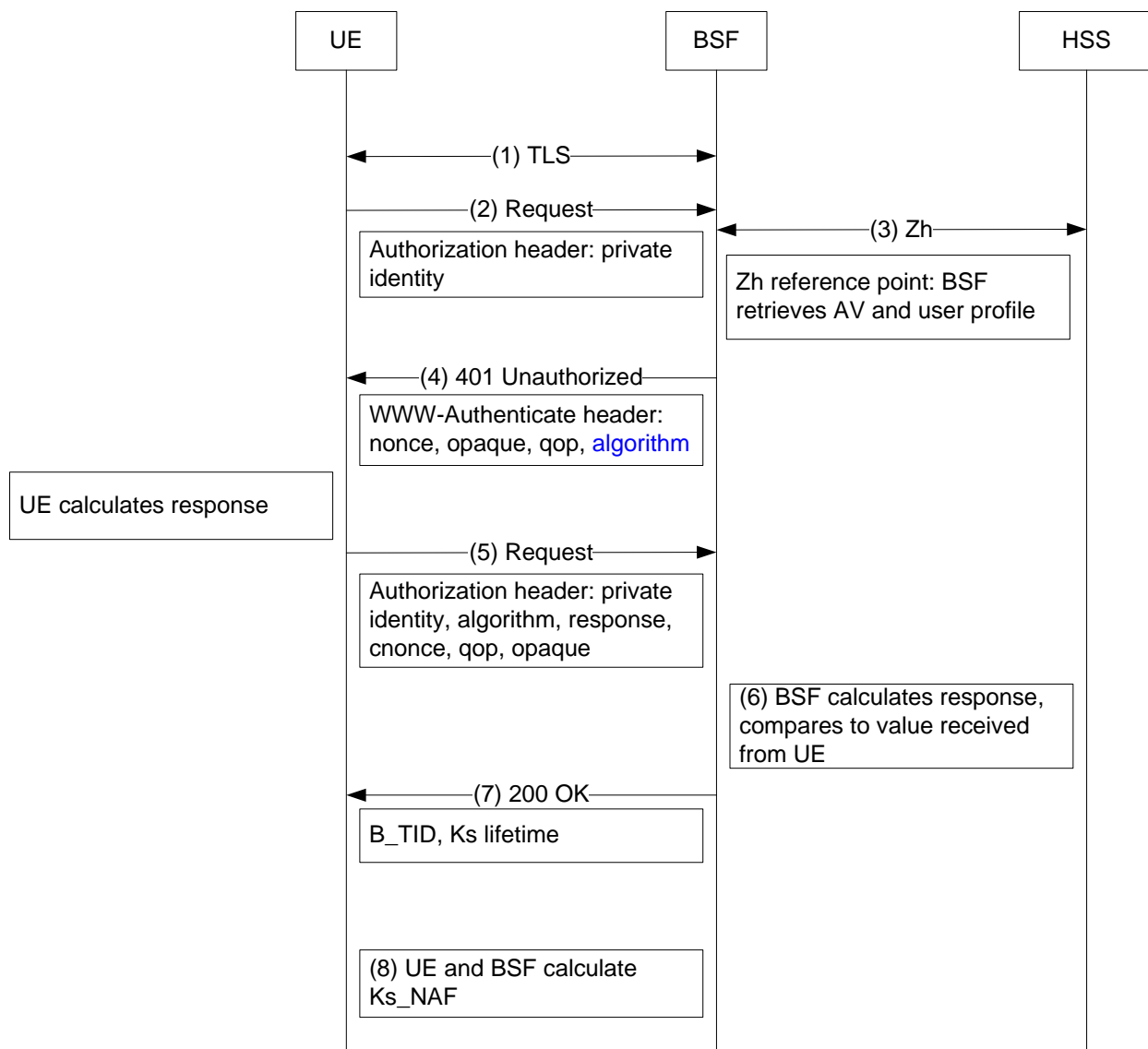


Figure 6.1: HTTP Digest Over TLS Bootstrapping Procedure

The new bootstrapping exchange on the Ub interface is illustrated in figure 6.1 and described below.

1. [The UE shall start the bootstrapping procedure by initiating a TLS session with the BSF. The UE and BSF shall negotiate server side authenticated TLS. The UE shall authenticate the BSF by the certificate presented by the BSF. The BSF does not require authentication from the UE at this point. REQ14836 The BSF certificate shall comply to the TLS Certificate Profile specified in Annex E of \[13\]. Information on the CableLabs certificate hierarchy can be found in Annex F of \[13\].](#)
2. [After negotiation of TLS, the UE shall send an HTTP Request message to the BSF containing the private identity in an Authorization header.](#)
3. [The BSF shall send a MAR command to the HSS to retrieve an authentication vector for that user. The HSS shall respond with the appropriate authentication vector for that user and algorithm in a MAA](#)

message. The authentication vector contents allow the BSF to calculate a challenge to the UE as described in RFC 2617 [3].

NOTE 1: In a multiple HSS environment, the BSF may have to obtain the address of the HSS where the subscription of the user is stored by querying the SLF, prior to step 3.

4. The BSF shall respond to the UE request with a 401 Unauthorized message containing a www-authenticate header to force the UE to authenticate itself. The www-authenticate header includes a 32 octet ASCII hexadecimal encoded nonce, created following the guidelines described in RFC 1750 [31]. The algorithm parameter informs the UE of the algorithm it should use to calculate its response.
5. Upon receiving the challenge, the UE shall use the data received in the www-authenticate header to create a second HTTP Request with the challenge response in an Authorization header. The challenge response is calculated per RFC 2617 [3]. A cnonce shall be included and calculated in the same manner as the nonce. The UE shall select a qop value from the list of qop values sent by the BSF and compute the response accordingly. The message shall be sent to the BSF over the TLS session.
6. The BSF shall check the validity of the challenge response sent by the UE by calculating the response on its own and comparing the values. The BSF calculates the response per RFC 2617 [3]. It uses the HA1 value supplied by the HSS over the Zh reference point.
7. If the challenge response sent by the UE is identical to the response calculated by the BSF, the BSF shall send a 200 OK message including the B-TID to the UE to indicate successful authentication. In addition, in a 200 OK message, the BSF shall supply the lifetime of the key Ks. The B-TID value shall be generated in the format of NAI by taking the base64 encoded [12] nonce value from step 4, and the BSF server name, i.e., base64encode(nonce)@BSF_servers_domain_name.

NOTE 2: Before base64 encoding the nonce from step 4, the nonce shall first be converted from a hexadecimal ASCII encoded value to a binary encoded value.

8. Both the UE and the BSF shall use the Ks to derive the key material Ks_NAF during the procedures as specified in clause 6.2. Ks_NAF shall be used for securing the reference point Ua.

Ks_NAF is computed as $Ks_NAF = KDF(Ks, "gba-h", nonce, IMPI, NAF_Id)$ where KDF is the key derivation function described in Annex B of 33.220. The binary encoded nonce is substituted for the AKA-based RAND variable when calculating Ks_NAF. Ks is the master secret from the existing TLS session.

The UE and the BSF shall store the key Ks with the associated B-TID for further use, until the lifetime of Ks has expired, or until the key Ks is updated. REQ14837 When the lifetime of the key Ks has expired, the UE shall terminate the protocol session over reference point Ua and perform a bootstrapping authentication over reference point Ub. The master secret of the TLS session established over the Ub interface between the UE and the BSF is the new key Ks. REQ14838 The default lifetime of the NAF specific key material Ks_NAF shall be equal to the lifetime of Ks when not specified within the Ua-application specification. REQ14839 The lifetime of the Ks_NAF shall not exceed the lifetime of the corresponding Ks.

6.2 Procedures Using Bootstrapped Security Association

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

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

1. UE starts communication over reference point Ua with the NAF:

- in general, UE and NAF will not yet share the key(s) required to protect the reference point Ua. If they already do (i.e., if a key Ks_NAF for the corresponding key derivation parameter NAF_Id is already available), the UE and the NAF can start to securely communicate right away. If the UE and the NAF do not yet share a key, the UE proceeds as follows:
 - if a key Ks is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause 6.1;
 - if no key Ks is available in the UE, the UE first agrees on a new key Ks with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF;

NOTE 1: If it is not desired by the UE to use the same Ks to derive more than one Ks_NAF then the UE should agree on a new key Ks with the BSF over the reference point Ub, and then proceed to derive Ks_NAF. The Ks key is initialized with the master secret of the TLS session. The implication of this procedure is that new Ks key can only be agreed on by the UE and the BSF if a new TLS session is setup and the challenge response procedure as described in clause 6.1 is executed once more.

- if the NAF shares a key with the UE, but the NAF requires an update of that key, e.g., because the key's lifetime has expired or will expire soon, or the key can not meet the NAF local validity condition, it shall send a suitable bootstrapping renegotiation request to the UE, see figure 4.5. If the key's lifetime has expired the protocol used over reference point Ua shall be terminated. The form of this indication depends on the particular protocol used over reference point Ua. If the UE receives a bootstrapping renegotiation request, it starts a run of the protocol over reference point Ub, as specified in clause 6.1, in order to obtain a new key Ks.

NOTE 2: To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see NOTE 2 of clause 6.1). For each protocol used over Ua it shall be specified if only cases (1) and (2) of NOTE 2 of clause 6.1 are allowed for the NAF or if the protocol used over Ua shall transfer also the FQDN used for key derivation by UE to NAF.

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

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

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

- key management for GBA related keys in the UE (i.e., Ks and Ks_NAF keys):
 - REQ10978 the Key Ks shall be deleted from the UE when the UE is powered down;
 - REQ11024 all other GBA related keys may be deleted from the UE when the UE is powered down. If the UE does not delete the GBA keys at power down then the GBA keys need to be stored in non-volatile memory.
 - REQ10979 when a new Ks is agreed over the reference point Ub and a key Ks_NAF, derived from one NAF_Id, is updated, the other keys Ks_NAF, derived from different values NAF_Id, stored on the UE shall not be affected;

NOTE 5: According to the procedures defined in clauses 6.1 and 6.2, in the UE there is at most one Ks_NAF key stored per NAF-Id.

2. NAF starts communication over reference point Zn with BSF

- The NAF requests key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua. REQ10980 If the NAF has several FQDNs, which may be used in conjunction with this

specification, then the NAF shall transfer in the request over Zn the same FQDN, which was used over Ua (see NOTE 2 on key derivation in this clause);

- The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;
- REQ10981 With the key material request, the NAF shall supply NAF's public hostname that UE has used to access NAF to BSF, and BSF shall be able verify that NAF is authorized to use that hostname;

3. The BSF derives the keys required to protect the protocol used over reference point Ua from the key Ks and the key derivation parameters, as specified in clause 6.1, and supplies to NAF the requested key Ks NAF, as well as the bootstrapping time and the lifetime of that key, and the requested application-specific and potentially NAF group specific USSs if they are available in subscriber's GUSS and if the NAF is authorized to receive the requested USSs. If the key identified by the B-TID supplied by the NAF is not available at the BSF, the BSF shall indicate this in the reply to the NAF. The NAF then indicates a bootstrapping renegotiation request to the UE.

NOTE 6: The NAF can further set the local validity condition of the Ks NAF according to the local policy, for example a limitation of reuse times of a Ks NAF.

NOTE 7: The NAF will adapt the key material Ks NAF to the specific needs of the reference point Ua in the same way as the UE did. This adaptation is outside the scope of this specification.

- The BSF may require that one or more application-specific and potentially NAF group specific USSs shall be present in subscriber's GUSS for the NAF (see clause 4.4.6).If one or more of these required settings are missing from the GUSS, the BSF shall indicate this in the reply to the NAF.
- The BSF may also send the private user identity (IMPI) and requested USSs to NAF according to the BSF's policy;

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

Once the run of the protocol used over reference point Ua is completed the purpose of bootstrapping is fulfilled as it enabled UE and NAF to use reference point Ua in a secure way. Details of messages exchanged between the UE and the NAF are described in [29].

Annex A: Void

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

B.1 Introduction

This annex specifies the key derivation function (KDF) that is used in the NAF specific key derivation in both GBA (i.e., GBA_ME) and GBA_U. The key derivation function defined in the annex takes the following assumptions:

1. the input parameters to the key derivation functions are octet strings - not bit strings of arbitrary length:
2. a single input parameter will have lengths no greater than 65535 octets.

B.2 Generic key derivation function

The input parameters and their lengths shall be concatenated into a string S as follows:

1. The length of each input parameter in octets shall be encoded into two-octet string:
 - a) express the number of octets in input parameter P_i as a number k in the range $[0, 65535]$.
 - b) L_i is then a two-octet representation of the number k , with the most significant bit of the first octet of L_i equal to the most significant bit of k , and the least significant bit of the second octet of L_i equal to the least significant bit of k ,

EXAMPLE: If P_i contains 258 octets then L_i will be the two-octet string 0x01 0x02.

2. String S shall be constructed from n input parameters as follows:

$$S = FC \parallel P_0 \parallel L_0 \parallel P_1 \parallel L_1 \parallel P_2 \parallel L_2 \parallel P_3 \parallel L_3 \parallel \dots \parallel P_n \parallel L_n$$

where

FC is single octet used to distinguish between different instances of the algorithm,

P_0 is a static ASCII-encoded string,

L_0 is the two octet representation of the length of the P_0 ,

$P_1 \dots P_n$ are the n input parameters, and

$L_1 \dots L_n$ are the two-octet representations of the corresponding input parameters.

3. The final output, i.e., the derived key is equal to HMAC-SHA-256 (as specified in [22] and [23]) computed on the string S using the key Key:

$$\text{derived key} = \text{HMAC-SHA-256}(\text{Key}, S)$$

B.2.1 Input parameter encoding

A character string shall be encoded to an octet string according to UTF-8 encoding rules as specified in IETF RFC 3629 [24].

B.3 NAF specific key derivation in GBA, ~~and~~ GBA_U, and GBA_H

In GBA, GBA_U, and GBA_H, the input parameters for the key derivation function shall be the following:

- FC = 0x01,
- P1 = RAND,
- L1 = length of RAND is 16 octets (i.e., 0x00 0x10),
- P2 = IMPI encoded to an octet string using UTF-8 encoding (see clause B.2.1),
- L2 = length of IMPI is variable (not greater than 65535),
- P3 = NAF_ID with the FQDN part of the NAF_ID encoded to an octet string using UTF-8 encoding (see clause B.2.1), and
- L3 = length of NAF_ID is variable (not greater than 65535).

In the key derivation of Ks_NAF as specified in clause 4 and Ks_ext_NAF as specified in clause 5,

- P0 = "gba-me" (i.e., 0x67 0x62 0x61 0x2d 0x6d 0x65), and
- L0 = length of P0 is 6 octets (i.e., 0x00 0x06).

In the key derivation of Ks_int_NAF as specified in clause 5,

- P0 = "gba-u" (i.e., 0x67 0x62 0x61 0x2d 0x75), and
- L0 = length of P0 is 5 octets (i.e., 0x00 0x05).

The Key to be used in key derivation shall be:

- Ks (i.e., CK || IK concatenated) as specified in clauses 4 and 5,

NOTE: In the specification this function is denoted as:

$Ks_NAF = KDF(Ks, "gba-me", RAND, IMPI, NAF_Id),$
 $Ks_ext_NAF = KDF(Ks, "gba-me", RAND, IMPI, NAF_Id),$ and
 $Ks_int_NAF = KDF(Ks, "gba-u", RAND, IMPI, NAF_Id).$

In GBA_H, the input parameters for the key derivation function shall be the following:

- REQ12727 FC = 0x01,
- P0 = "gba_h" (i.e., 0x67 0x62 0x61 0x2d 0x68),
- L0 = length of P0 is 5 octets (i.e., 0x00 0x06)
- REQ12728 P1 = nonce,
- L1 = length of nonce is 16 octets (i.e., 0x00 0x10),
- P2 = IMPI encoded to an octet string using UTF-8 encoding (see clause B.2.1),
- L2 = length of IMPI is variable (not greater than 65535),
- P3 = NAF_ID with the FQDN part of the NAF_ID encoded to an octet string using UTF-8 encoding (see clause B.2.1), and

- L3 = length of NAF_ID is variable (not greater than 65535).

The Key to be used in key derivation shall be:

- Ks (i.e., the TLS session master secret) as specified in clause 6.1.

NOTE: In the specification this function is denoted as:

$Ks_{NAF} = KDF(Ks, "gba-h", nonce, IMPI, NAF_Id).$

Annex C (informative): Void

Annex D (informative): Dialog example for user selection of UICC application used in GBA

For certain cases, clause 4.4.8 specifies user involvement in the selection of the UICC application used for GBA procedures. A dialog window example for such an involvement is described below:

- The title of the dialog: "Authentication request".
- Explanation: "A service requires you to authenticate, please select your identity:"
- List of identities: A selectable list of applications on the UICC. The text visible for each application is extracted from the "Label" field of the application list on the UICC.
- Buttons: "Select" and "Cancel".

Annex E (normative): TLS profile for securing Zn' reference point

The TLS profile for securing the Zn' reference point is specified in RFC 3588 [14] section 13.2.

In addition, the D-Proxy certificate, i.e., the client certificate used in TLS handshake shall contain the subjectAltName extension as specified in RFC 3280 [17]. The subjectAltName extension shall contain one or more dNSName names. The dNSName name may contain the wildcard character '*' and the matching is performed as specified in RFC 2818 [18] section 3.1.

NOTE: The D-Proxy certificate shall contain all the DNS names of NAFs that may send a request for NAF specific shared secret through the D-Proxy to the subscriber's home BSF. If a new NAF is added, the new DNS name is either covered in the certificate by using the wildcard character approach (e.g., "*.operator.com"), or a new dNSName name needs to be added to the certificate. In the latter case, new certificate is needed for the D-Proxy.

Annex F (informative): Handling of TLS certificates

An authentication framework as available for IPsec [19] is not available for TLS certificates. The purpose of this Annex is to provide guidelines for TLS certificate handling for use on the Zn' reference point in the absence of a framework for TLS certificates.

Within this Annex following abbreviations are used: CA_A is the certification authority in A's network and CA_B is the certification authority in B's network. $Cert_A$ is the certificate of A and $Cert_B$ is the certificate of B. I_A is the set of identifiers that A may use to identify the NAF towards the BSF. T_B is the set of peers trusted by B.

F.1 TLS certificate enrollment

Mutual authentication in TLS is achieved based on public key technology and certificates. Both TLS peers A and B need to contain a certificate store and there shall be at least one certification authority CA that can issue certificates within the security domains in [with-which](#) A and B are [a](#) part of. $Cert_A$ contains the set I_A of A's identifiers. Each identifier is in the form of fully qualified domain name (FQDN). Similarly, B's certificate is $Cert_B$.

The certificates in the store of B define the group T_B of peers trusted by B. There are several options for creation and enrollment of certificates, three of which are described below.

1. In one option there is a certification authority, CA_B , only in the network of B. CA_B issues a certificate $Cert_B$ to B and a certificate $Cert_A$ to A. The certificates are delivered from CA_B to A and B in a secure way "out of band". Both A and B then add their peer into the group of their trusted peers by inserting that peer's certificate into the certificate store: A inserts $Cert_B$ into A's certificate store and B inserts $Cert_A$ into B's certificate store. This insertion is typically manual and the details depend on the implementation of the management interface to the certificate store.
2. In another option both A's and B's networks contain certification authorities, CA_B and CA_A , respectively. CA_B issues a certificate $Cert_B$ to B and CA_A issues a certificate $Cert_A$ to A. The certificates are delivered from CA_B to A and from CA_A to B in a secure way "out of band". Both A and B then add their peer into the group of their trusted peers by inserting that peer's certificate into the certificate store: A inserts $Cert_B$ into A's certificate store and B inserts $Cert_A$ into B's certificate store.
3. In a third option the CA certificates of both sides are exchanged: the certificate of CA_B is delivered to A and the certificate of CA_A is delivered to B in a secure way "out of band", inserted to the certificate store, and marked trusted. The validation of $Cert_A$ and $Cert_B$, that are exchanged during TLS handshake, is based on the presence of the corresponding CA certificates in the certificate store.

NOTE: In options 1 and 2 the need for certification authority may be avoided if the peers generate self signed certificates and exchange them in a secure way, "out of band". Also, instead of certificates themselves, certificate fingerprints may be exchanged "out of band" in those options.

F.2 TLS Certificate revocation

In the absence of PKI-revocation interfaces, certificate revocation needs to be performed manually. The revocation operation involves the removal of A from the group T_B of peers trusted by B. In the first two enrollment options described above the revocation happens by B removing the certificate of A, $Cert_A$, from its certificate store. This removal can be done manually. In the third option the certificate of A, $Cert_A$, is not in B's certificate store. For that reason B has to have a way to check the validity of $Cert_A$ with the issuer of the certificate (also in the first two

enrollment options the amount of manual maintenance operations will decrease if B can check the validity of Cert_A with the issuer of the certificate). This check may be done by using Online Certificate Status Protocol (OCSP) [20] or by using Certificate Revocation Lists (CRLs) [17] published by the issuer of CertA.

Annex G (normative): GBA_U UICC-ME interface

This annex describes the UICC-ME interface to be used when a GBA_U aware UICC application is active and the ME is involved in a GBA bootstrapping procedure. When the UICC application is not GBA_U aware, the ME uses AUTHENTICATE command in non-GBA_U security context (i.e., UMTS security context in case of USIM application and IMS security context in case of the ISIM) as defined in TS 31.102 [1] and TS 31.103 [10].

G.1 GBA_U Bootstrapping procedure

This procedure is part of the Bootstrapping procedure as described in clause 5.3.2.

The ME sends RAND and AUTN to the UICC, which performs the Ks derivation as described in clause 5.3.2.

The UICC then stores Ks. The UICC also stores the used RAND to identify the current bootstrapped values. RAND value in the UICC shall be further accessible by the ME.

The ME then finalizes the Bootstrapping procedure and stores in the UICC the Transaction Identifier (B-TID) and Key Life Time associated with the previous bootstrapped keys (i.e., Ks). Transaction Identifier and Key Life Time values in the UICC shall be further accessible by the ME.

At the end of the GBA_U bootstrapping procedure the UICC stores Ks, Transaction Identifier, Key Life Time and the RAND.

The UICC sends RES to the ME.

A new bootstrapping procedure replaces Ks, B-TID, Key LifeTime and RAND values of the previous bootstrapping procedure.

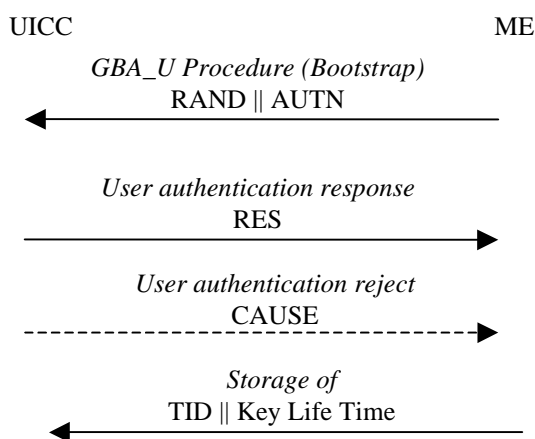


Figure G.1: GBA_U Bootstrap Procedure

G.2 GBA_U NAF Derivation procedure

This procedure is part of the Procedures using bootstrapped Security Association as described in clause 5.3.3

The ME sends NAF_ID and IMPI to the UICC. The UICC then performs Ks_ext_NAF and Ks_int_NAF derivation as described in clause 5.3.2. The UICC uses the RAND and Ks values stored from the previous bootstrapping procedure. The UICC returns Ks_ext_NAF to the ME and stores Ks_int_NAF and associated B-TID together with NAF_Id.

NOTE: A previous GBA_U Bootstrap needs to be undertaken before. If Ks is not available in the UICC, the command will answer with the appropriate error message.

The input parameters IMPI and the FQDN part of NAF_ID shall be encoded to octet strings using UTF-8 encoding rules as specified in IETF RFC 3629 [24].

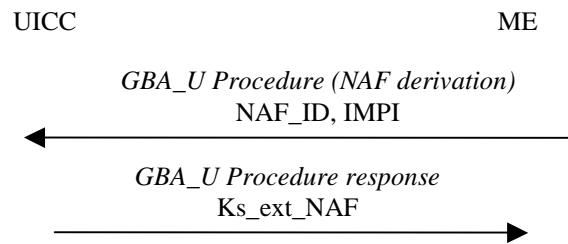


Figure G.2: GBA_U NAF derivation procedure

Annex H (normative): Ua security protocol identifier

H.1 Definition

The Ua security protocol identifier is a string of five octets. The first octet denotes the organization which specifies the Ua security protocol. The four remaining octets denote a specific security protocol within the responsibility of the organization.

For all Ua protocols specified by 3GPP this Annex shall contain a complete list of these protocols. For Ua protocols specified by other organizations this Annex shall only specify the organization octet of the Ua security protocol identifier. Two organization octets are reserved for special use.

H.2 Organization Octet

The organization octet denotes the organization specifying the particular protocol. Each organization intending to specify a Ua security protocol identifier shall apply to 3GPP to receive an organization octet value, which shall be registered within this Annex. Following is a list of registered organization octets:

“0x00” as first octet is the default value for protocols not specified otherwise. When octet “0x00” is used as first octet, only Ua security protocol identifier (0x00,0x00,0x00,0x00,0x00) shall be used.

NOTE: All protocols having this Ua security protocol identifier cannot be separated from each other.

“0x01” .. “0xFE” as the first octet denote organizations specifying Ua security protocol identifiers.

“0xFF” as the first octet denotes the private range of Ua security protocol identifiers.

NOTE: identifiers with “0xFF” as first octet may be used for defining local/experimental protocols without need for registration. When using such an identifier, however, it may happen that a security breach in one security protocol over Ua can be exploited by an attacker to mount successful attacks on a different security protocol over Ua.

The following values for organizations are assigned:

“0x01” 3GPP

NOTE: All protocols having the organization octet “0x01” are specified in annex H.3.

“0x02” 3GPP2

“0x03” Open Mobile Alliance

“0x04” GSMA

H.3 Ua security protocol identifiers for 3GPP specified protocols

The following Ua security protocol identifiers are specified by 3GPP:

(0x01,0x00,0x00,0x00,0x00) Ua security protocol according to TS 33.221 [5].

(0x01,0x00,0x00, 0x00,0x01) Ua security protocols according to TS 33.246 [26].NOTE: TS 33.246[26] provides key separation between the keys that are used within HTTP digest and MIKEY protocols.

(0x01,0x00,0x00, 0x00,0x02) Ua security protocol HTTP digest authentication according to TS 24.109 [29], unless HTTP digest authentication is used in the context of another Ua security protocol, which is already covered elsewhere in this Annex.

(0x01,0x00,0x01,yy,zz) Ua security protocol for “Shared key-based UE authentication with certificate-based NAF authentication”, according to TS 33.222 [25], or “Shared key-based mutual authentication between UE and NAF”, according to TS 33.222 [25]. Here, “yy,zz” is the protection mechanism CipherSuite code according to the defined values for TLS CipherSuites in TLS V1.0 [28] and PSK Ciphersuites for TLS [27].

NOTE: As an example: RFC2246 [28] CipherSuite TLS_RSA_WITH_3DES_EDE_CBC_SHA has code = { 0x00,0x0A }, thus the according protocol identifier shall be (0x01,0x00,0x01,0x00,0x0A).

Appendix I CableLabs Acknowledgements

We wish to thank the vendor participants contributing directly to this document:

Sorin Georgescu - Ericsson

Wassim Haddad - Ericsson

Louis LeVay - Nortel

Steve Dotson - CableLabs

Appendix II Change History

Base document for I01:

3GPP TS 33.220 V6.7.0 (2005-12) plus cable-specific changes.

Base document for I02:

3GPP TS 33.220 V6.7.0 (2005-12) plus cable-specific changes and the following engineering changes.

<u>ECN</u>	<u>ECN Date</u>	<u>Summary</u>
<u>33.220-N-06.0344-3</u>	<u>9/11/06</u>	<u>Clarifications for Ks NAF, Reference to 33.210 Appendix, Minor Technical and Editorial Changes</u>