ARIB STD-T63-33.220 V11.5.0

Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)

(Release 11)

Refer to “Industrial Property Rights (IPR)” in the preface of ARIB STD-T63 for Related Industrial Property Rights. Refer to “Notice” in the preface of ARIB STD-T63 for Copyrights.

3GPP TS 33.220 V11.5.0 (2013-06)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Generic Authentication Architecture (GAA);
Generic Bootstrapping Architecture (GBA)
(Release 11)

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organisational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organisational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organisational Partners' Publications Offices.

Keywords

LTE, GSM, UMTS, access, security, IP,
Multimedia, SIP

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.

© 2013, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).
All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners
LTE™ is a Trade Mark of ETSI currently being registered for the benefit of its Members and of the 3GPP Organizational Partners
GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword 7
1 Scope 8
2 References 8
3 Definitions, abbreviations symbols and conventions 10
3.1 Definitions 10
3.2 Abbreviations 11
3.3 Symbols 11
3.4 Conventions 11
4 Generic Bootstrapping Architecture 12
4.1 Reference model 12
4.2 Network elements 14
4.2.1 Bootstrapping server function (BSF) 14
4.2.2 Network application function (NAF) 14
4.2.2a Zn-Proxy 15
4.2.3 HSS 15
4.2.4 UE 16
4.2.5 SLF 16
4.2.6 HLR 17
4.3 Bootstrapping architecture and reference points 17
4.3.1 Reference point Ub 17
4.3.2 Reference point Ua 17
4.3.3 Reference point Zh 17
4.3.4 Reference point Zn 17
4.3.5 Reference point Dz 17
4.3.6 Reference point Zh' 17
4.4 Requirements and principles for bootstrapping 17
4.4.1 Access Independence 18
4.4.2 Authentication methods 18
4.4.3 Roaming 18
4.4.4 Requirements on reference point Ub 18
4.4.5 Requirements on reference point Zh 19
4.4.6 Requirements on reference point Zn 19
4.4.7 Requirements on Bootstrapping Transaction Identifier 20
4.4.8 Requirements on selection of UICC application and related keys 21
4.4.8.1 UICC application activation procedure in GBA 22
4.4.9 Requirements on reference point Ua 23
4.4.10 Requirements on reference point Dz 23
4.4.11 Requirements on GBA keys and parameters handling 23
4.4.12 Requirements on reference point Zh' 24
4.4.13 Requirements on TMPI handling 25
4.5 Procedures 25
4.5.1 Initiation of bootstrapping 25
4.5.2 Bootstrapping procedures 25
4.5.3 Procedures using bootstrapped Security Association 28
4.5.4 Procedure related to service discovery 30
5 UICC-based enhancements to Generic Bootstrapping Architecture (GBA_U) 30
5.1 Architecture and reference points for bootstrapping with UICC-based enhancements 31
5.2 Requirements and principles for bootstrapping with UICC-based enhancements 31
5.2.1 Requirements on UE 31
5.2.2 Requirements on BSF 31
5.3 Procedures for bootstrapping with UICC-based enhancements 31
5.3.1 Initiation of bootstrapping 31
5.3.2 Bootstrapping procedure 31
5.3.3 Procedures using bootstrapped Security Association 34

5.3.4 Procedure related to service discovery 36

Annex A: (Void) 37

Annex B (normative): Specification of the key derivation function KDF 38

B.1 Introduction 38

B.2 Generic key derivation function 38
B.2.1 Input parameter encoding 38
B.2.1.1 General 38
B.2.1.2 Character string encoding 39
B.2.1.3 Non-negative integer encoding 39
B.2.2 FC value allocations 39

B.3 NAF specific key derivation in GBA and GBA_U 39

B.4 Derivation of TMPI 40

B.5 Derivation of passwd and Ks 41

B.6 NAF specific key derivation in GBA_Digest 41

Annex C: (Void) 42

Annex D (informative): Dialog example for user selection of UICC application used in GBA 43

Annex E (normative): TLS profile for securing Zn/Zn' reference points 44

Annex F (informative): Handling of TLS certificates 45

Annex G (normative): GBA_U UICC-ME interface 46

G.1 GBA_U Bootstrapping procedure 46

G.2 GBA_U NAF Derivation procedure 46

Annex H (normative): Ua security protocol identifier 48

H.1 Definition 48

H.2 Organization Octet 48

H.3 Ua security protocol identifiers for 3GPP specified protocols 48

Annex I (normative): 2G GBA 50

I.0 Introduction 50

I.1 Reference model 50

I.2 Network elements 50
I.2.1 Bootstrapping server function (BSF) 50
I.2.2 Network application function (NAF) 51
I.2.2a Zn-Proxy 51
I.2.3 HSS 51
I.2.4 UE 52
I.2.5 SLF 52
I.2.6 HLR 52

I.3 Bootstrapping architecture and reference points 53
I.3.1 Reference point Ub 53
I.3.2 Reference point Ua 53
I.3.3 Reference point Zh 53
I.3.4 Reference point Zn 53
I.3.5 Reference point Dz 53
I.3.6 Reference point Zh' 53

I.4 Requirements and principles for bootstrapping....53
I.4.0 General requirements....53
I.4.1 Access Independence....54
I.4.2 Authentication methods....54
I.4.3 Roaming....54
I.4.4 Requirements on reference point Ub....54
I.4.5 Requirements on reference point Zh....54
I.4.6 Requirements on reference point Zn....55
I.4.7 Requirements on Bootstrapping Transaction Identifier....56
I.4.8 Requirements on selection of UICC application and SIM card....57
I.4.9 Requirements on reference point Ua....57
I.4.10 Requirements on reference point Dz....57
I.4.11 Requirements on reference point Zh'....57

I.5 Procedures....57
I.5.1 Initiation of bootstrapping....57
I.5.2 Bootstrapping procedures....58
I.5.3 Procedures using bootstrapped Security Association....61
I.5.4 Procedure related to service discovery....63

I.6 TLS Profile....63
I.6.1 void....64
I.6.2 Authentication of the BSF....64
I.6.3 Authentication of the UE....64
I.6.4 Set-up of Security parameters....64

Annex J (informative): Usage of USS with local policy enforcement in BSF....65

J.1 General....65

J.2 Usage scenarios....65
J.2.1 Scenario 1: NAF does not use USSs, BSF does not have local policy for NAF....66
J.2.2 Scenario 2: NAF does not use USSs, BSF does have local policy for NAF....66
J.2.3 Scenario 3: NAF does use USSs, BSF does not have local policy for NAF....66
J.2.4 Scenario 4: NAF does use USSs, BSF does have local policy for NAF....67

Annex K (informative): Interoperator GBA-usage examples....68

K.1 Example on interoperator GBA setup....68

K.2 Example on interoperator GBA operation....70

Annex L (informative): Information on how security threats related to known GSM vulnerabilities are addressed by the 2G GBA solution....73

L.1 Impersonation of the UE to the BSF during the run of the Ub protocol....73

L.2 Impersonation of the BSF to the UE during the run of the Ub protocol....73

L.3 Finding the GBA key Ks during or after the Ub protocol run....74

L.4 Bidding down attack....74

Annex M (normative): GBA_Digest....74

M.1 General....74

M.2 Reference model....74

M.3 Network elements....74
M.3.1 Bootstrapping server function (BSF)....74
M.3.2 Network application function (NAF)....75
M.3.3 Zn-Proxy....76
M.3.4 HSS....76
M.3.5 UE....76
M.3.6 SLF....77

M.4 Bootstrapping architecture and reference points 77
M.4.1 Reference point Ub 77
M.4.2 Reference point Ua 77
M.4.3 Reference point Zh 77
M.4.4 Reference point Zn 77
M.4.5 Reference point Dz 77

M.5 Requirements and principles for bootstrapping 77
M.5.1 General Requirements 77
M.5.2 Access independence 78
M.5.3 Authentication methods 78
M.5.4 Roaming 78
M.5.5 Requirements on reference point Ub 78
M.5.6 Requirements on reference point Zh 79
M.5.7 Requirements on reference point Zn 79
M.5.8 Requirements on Bootstrapping Transaction Identifier 81
M.5.9 Requirements on reference point Ua 81
M.5.10 Requirements on reference point Dz 81
M.5.11 Requirements on GBA keys and parameters handling 81

M.6 Procedures 81
M.6.1 General 81
M.6.2 Initiation of bootstrapping 81
M.6.3 Bootstrapping procedures 82
M.6.4 Procedures using bootstrapped Security Association 84
M.6.5 Procedure related to service discovery 87

M.7 TLS Profile 87
M.7.1 General 87
M.7.2 Authentication of the BSF 87
M.7.3 Authentication of the UE 88
M.7.4 Set-up of Security parameters 88

Annex N (informative): Change history 89

Foreword

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

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

Version x.y.z

where:

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

1 Scope

The present document describes the security features and mechanisms to bootstrap authentication and key agreement for application security. 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 generic bootstrapping functions, an architecture overview and the detailed procedure 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. Annex I of this specification describes a mechanism, called 2G GBA, to bootstrap authentication and key agreement using 2G AKA protocol. Annex M of this specification describes a mechanism, called GBA_Digest, to bootstrap authentication and key agreement using HTTP Digest protocol with SIP Digest credentials.

2 References

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

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

[1] 3GPP TS 31.102: "Characteristics of the USIM application".

[2] 3GPP TS 33.102: "3G Security; Security architecture".

[3] IETF RFC 2617: "HTTP Authentication: Basic and Digest Access Authentication".

[4] IETF RFC 3310: "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)".

[5] 3GPP TS 33.221: "Generic Authentication Architecture (GAA); Support for Subscriber Certificates".

[6] Void

[7] Void

[8] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2 (Release 6)".

[9] Void.

[10] 3GPP TS 31.103: "Characteristics of the IP Multimedia Services Identity Module (ISIM) application".

[11] 3GPP TS 23.003: "Numbering, addressing and identification".

[12] IETF RFC 3548: "The Base16, Base32, and Base64 Data Encodings".

[13] 3GPP TS 33.210: "3G Security; Network domain security; IP network layer security".

[14] Void.

[15] 3GPP TS 31.101: "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] Void.

[18] IETF RFC 2818: "HTTP over TLS".

[19] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[20] Void.

[21] Void.

[22] IETF RFC 2104: "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: "UTF-8, a transformation format of ISO 10646".

[25] 3GPP TS 33.222: "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] Void.

[28] IETF RFC 2246: "The TLS Protocol Version 1".

[29] 3GPP TS 24.109: "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details".

[30] (void)

[31] (void)

[32] 3GPP TS 29.109: "Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on the Diameter protocol; Stage 3".

[33] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

[34] 3GPP TS 23.002: “Network architecture “.

[35] 3GPP TS 33.401: "3GPP System Architecture Evolution (SAE); Security Architecture".

[36] 3GPP TS 33.402: "3GPP System Architecture Evolution (SAE); Security aspects of non-3GPP accesses".

[37] "Unicode Standard Annex #15; Unicode Normalization Forms", Unicode 5.1.0, March 2008. http://www.unicode.org

[38] 3GPP TS 26.237: "IP Multimedia Subsystem (IMS) based Packet Switch Streaming (PSS) and Multimedia Broadcast/Multicast Service (MBMS) User Service; Protocols".

[39] 3GPP TS 33.224: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push Layer".

[40] 3GPP TS 33.328: "IMS Media plane security".

[41] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[42] (void)

[43] Void.

[44] IETF RFC 5705: "Keying Material Exporters for Transport Layer Security (TLS)".

[45] 3GPP TS 33.223: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) Push function".

3 Definitions, abbreviations symbols and conventions

3.1 Definitions

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

AKA-based GBA: This term collectively refers to all GBA variants that make use of a form of the AKA protocol on the Ub interface, i.e. the term refers to GBA_ME, GBA_U, and 2G GBA, as defined in the present document, and to GBA_push as defined in TS 33.223 [45].

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.

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.

GBA_Digest: A GBA variant that extends the usage of GBA to environments where the UICC is not available to the subscriber. In this variant, the GBA client on the UE and the BSF communicate using HTTP protocol and SIP Digest credentials, such as a shared secret or password, that are used for authentication instead of credentials stored in the SIM, USIM or ISIM.

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.

GUSS timestamp: the timestamp of the GUSS is set by the HSS. It changes whenever the HSS has modified the GUSS.

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 FQDN of the NAF, concatenated with the Ua security protocol identifier.

Temporary IP Multimedia Private Identity: a temporary identity which is used on the Ub interface to prevent passive eavesdropping attacks against the IMPI.

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). In addition, a USS may contain a key selection indication, which is used in the GBA_U case to mandate the usage of either the ME-based key (Ks_(ext)_NAF) or the UICC-based key (Ks_int_NAF) or both. 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_ME	ME-based GBA
GBA_U	GBA with UICC-based enhancements
GUSS	GBA User Security Settings
HLR	Home Location Register
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
TMPI	Temporary IP Multimedia Private Identity
USS	User Security Setting

3.3 Symbols

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

$\|\|$	Concatenation
$\oplus$	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

For HLR and HSS definitions used in this chapter refer to [34].

When HSS is mentioned in this specification without an indication of supported reference point towards the BSF, then the support of the Zh reference point is meant.

Figure 4.1 shows a simple network model of the entities involved in the bootstrapping approach when an HSS with Zh reference point is deployed, and the reference points used between them.

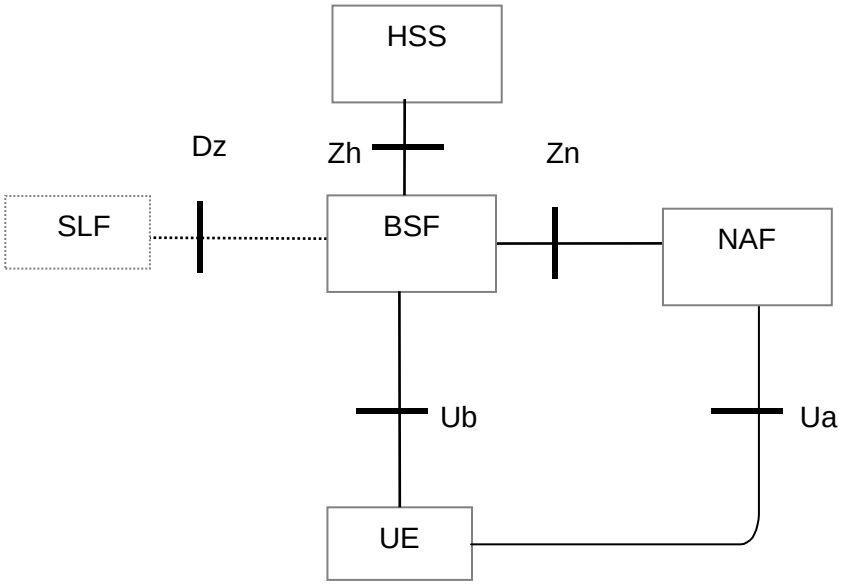
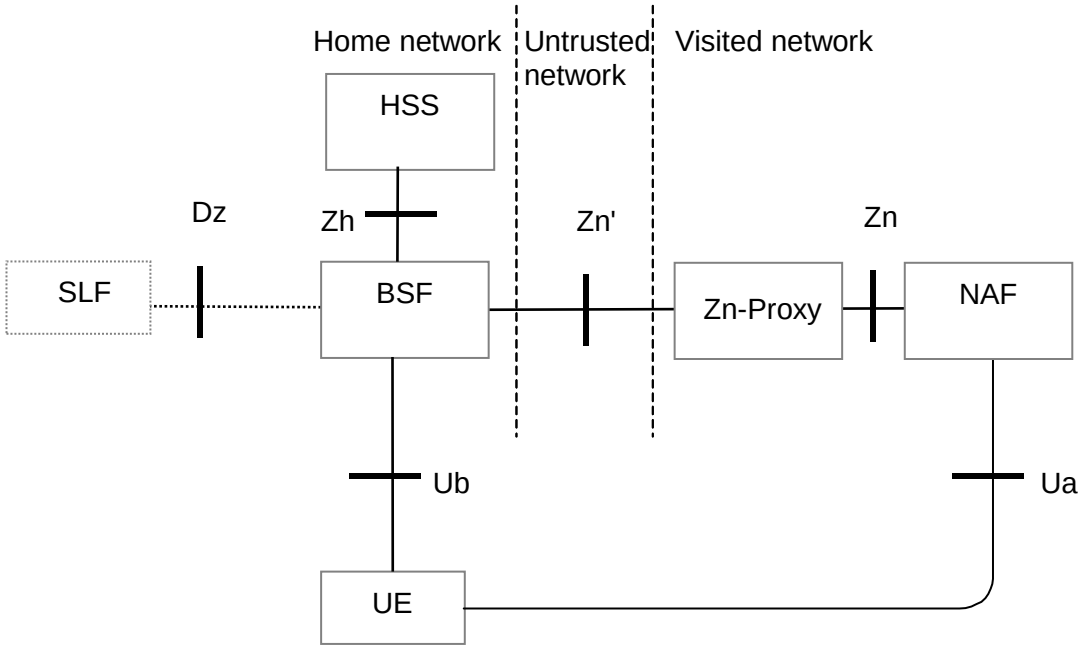


Figure 4.1: Simple network model for bootstrapping involving HSS with Zh reference point

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 involving HSS with Zh reference point

Figure 4.1b shows a simple network model of the entities involved in the bootstrapping approach when either an HLR or an HSS without Zh reference point support is deployed, and the reference points used between them. The reference point Zh' is optional for the BSF to support.

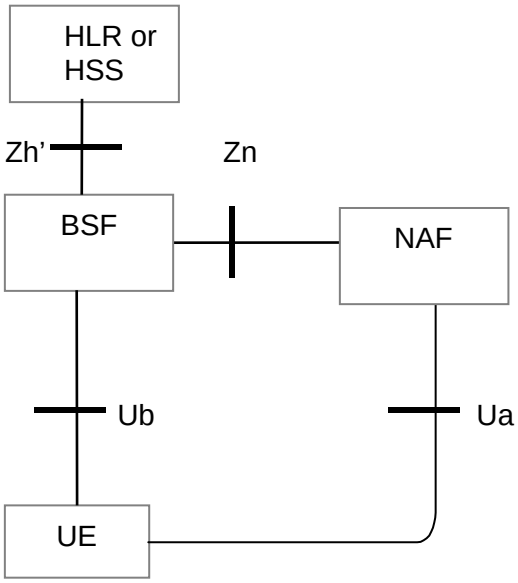


Figure 4.1b: Simple network model for bootstrapping involving either an HLR or an HSS without Zh reference point support

4.2 Network elements

4.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the AKA protocol, and agree on session keys that are afterwards applied between UE and 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.

NOTE 3: If support of GBA User Security Settings (GUSS) for service differentiation or GBA_U is desired in combination with HLR or HSS without Zh reference point support, then this can be achieved, for instance by storing the GUSS information in a BSF database (external and/or external to the node itself), or in any other network database which is deemed as appropriate for a specific deployment. GUSS information is not sent over Zh' reference point.

If an HLR or an HSS without Zh reference point support is used within the GBA architecture, then the BSF needs to be configured to use the Zh' reference point with that HLR or HSS. If the Zh reference point is available in the HSS and the full migration has happened, then it shall be used between the BSF and the HSS.

NOTE 4: If an operator wants to upgrade from a GBA architecture using HLR or HSS without Zh reference point support, to one using HSS with Zh reference point support, then the BSF needs to be configured accordingly to use then the Zh reference point. This can also involve a configuration, where gradual replacement is needed. If GBA is deployed from the beginning with an HSS with Zh reference point support then this kind of configuration is not needed.

NOTE 5: During migration from HLR to HSS, the BSF will need to select for a subscriber between HSS and HLR's. Such a mechanism (e.g. configuration based) will not be standardized.

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;

- in the case of GBA_U, the NAF shall be able to determine which key (i.e., Ks_ext_NAF or Ks_int_NAF or both) should be used by using a local policy in the NAF or a key selection indication in the application-specific USS. If the NAF has received an application-specific USS, which contains the key selection indication, this shall override the local policy in the NAF;

- 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(_int/ext)_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(_int/ext)_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 Zn-Proxy

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

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

General requirements for the functionality of Zn-Proxy are:

- Zn-Proxy shall be able to function as a proxy between the visited NAF, and the subscriber's home BSF;

- Zn-Proxy shall be able to locate subscriber's home BSF and communicate with it over secure channel;

- Zn-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 Zn-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 Zn-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 may contain one or more GUSSs that can be mapped to one or more private identities, i.e. IMPIs and IMSIs. Each of the existing GUSSs shall be mapped to one or more private identities, but each private identity shall only have zero or one GUSS mapped to it.

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 key selection indication in the case of GBA_U (i.e., whether the NAF shall use Ks_ext_NAF or Ks_int_NAF), identification or authorization information of one or more applications hosted by one ore 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 from its local database without involvement with the HSS.

NOTE 2: One possibility to revoke temporarily an application specific USS from the GUSS is that the HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber. The GUSS in the BSF is not changed by this operation and only updated when the existing bootstrapping session times out, or is overwritten by a new bootstrapping session during which the new modified GUSS is fetched from HSS along with the AV.

- 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:
  - optionally the timestamp indicating the time when the GUSS has been last modified by the HSS.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or 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 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]).

A GBA-aware ME 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.

4.2.6 HLR

If a HLR is used, then the requirement on the HLR is:

- The HLR shall support the request from the BSF for the required authentication vector.

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.

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].

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 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 reference point Zh is an intra-operator domain interface. 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 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.

4.3.6 Reference point Zh'

The reference point Zh' used between the BSF and the HLR allows the BSF to fetch the required authentication information. The reference point Zh' is an intra-operator domain interface.

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.

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

- the BSF and the UE shall protect the permanent user identity IMPI against passive eavesdropping attacks by using a temporary identity. The support of the temporary identity by UE or BSF shall not preclude a successful bootstrapping procedure if the other entity conforms to an earlier release of this specification and does not support the use of a temporary identity.

NOTE 2: User identity privacy can be achieved only when both, UE and BSF, support the use of a temporary identity.

NOTE 3: The use of a temporary identity is not required for 2G GBA (cf. Annex I) as the IMPI is already protected by the mandatory TLS tunnel.

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 since BSF and HSS are located within the same operator's network.

- the BSF shall be able to send bootstrapping information request concerning a subscriber;
- optionally the BSF may have the capability able to send the timestamp of subscriber's GBA user security settings to the HSS (timestamp option);
- the HSS shall be able to send one 3GPP AKA vector at a time 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. Optionally the HSS may have the capability to indicate to the BSF whether the BSF already has the latest copy of the GUSS based on the GUSS timestamp (timestamp option);

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 DIAMETER based Zn reference point shall be secured according to NDS/IP [13] or may be secured using TLS as specified in Annex E of the present document;
- If the BSF and the NAF are located in different operators' networks, the DIAMETER based Zn' reference point between the Zn-Proxy and the BSF shall be secured using TLS as specified in Annex E of the present document;
- An HTTP based Zn/Zn' reference point shall be secured using TLS as specified in Annex E of the present document;
- 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 or MSISDN, 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
- whether private subscriber identity, i.e. IMPI, may be sent to the NAF;

- whether a particular USS may be sent to a NAF;

NOTE 3: Privacy issues need be considered when determining which user identifier is sent to the NAF. If service continuity is desired, then the BSF can be configured to send the IMPI. If HLR is utilized instead of HSS, BSF can be configured to send MSISDN over Zn (but then there is no user anonymity). If the BSF does not send the IMPI, MSISDN or IMPU / pseudonym in the USS, then the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user identifier. This can cause that the NAF cannot provide a continuous service, since a user identity is needed in the NAF to ensure that the NAF is able to update keys for a Ua session when the UE has bootstrapped and contacts the NAF with a new B-TID. If user privacy is desired, the NAF can requests a USS and the BSF is configured to send a user pseudonym in the USS, but not the IMPI.

- 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;

NOTE 4: For more information on the local policy usage, see Annex J.

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

NOTE 6: 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).

- The BSF shall remove any existing attribute indicating NAF Grouping from the USSs sent to NAFs.

- NAF shall be able to indicate to BSF the protocol identifier of Ua security protocol it requires the key material by sending NAF-Id to BSF (cf. Annex H).

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

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.

NOTE 1: A Ua application specification may require the use of only a USIM (e.g. in MBMS) or only an ISIM.

NOTE 2: A user or operator may want to use a Ua application with a specific UICC application indicated by the “Label”. This could be configured in the Ua application in the ME by the user or the operator.

A Ua application may require to use the same UICC application in the first and all consecutive runs of Ub protocol for a Ua application instance to ensure that IMPI is not changed during a Ua application session which lasts over several runs of Ub protocol. In this case the Ua application shall request the GBA function to run the Ub protocol with the UICC application that is indicated by the corresponding "Label" or IMPI, depending on which one is available. If both are available, then IMPI shall be used to indicate which UICC application is to be used by the GBA function.

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

At most, there can be only one USIM active at one time. Therefore, if the USIM indicated in the "Label" by the Ua application is different to the currently active USIM application, then the ME shall reject this request.

If a different ISIM to the currently active ISIM application(s) is indicated to the GBA support function by the Ua application, then the ME shall not terminate the currently active ISIM application(s) but the ME shall follow the procedure in chapter 4.4.8.1 when activating the ISIM application indicated by the "Label", as the UE is allowed to have several ISIM's active simultaneously.

   c. if no "Label" was indicated in step a and there are UICC applications active:

If a preferred UICC application type was indicated but no UICC application of this type is active then step d shall be followed.

If a preferred UICC application type was indicated and there are active UICC applications of this preferred type, then the GBA function shall choose:

- if the preferred UICC application type is USIM then the active USIM is selected

- if the preferred UICC application type is ISIM and only one ISIM is active then this is selected

- if the preferred UICC application type is ISIM and more than one ISIM is active then the GBA function may show a UICC application choosing dialogue to the end user (the list contains the "Labels" from the application list of all active ISIM applications on the UICC), from which the end user chooses the UICC application to be selected; if no dialogue is shown the GBA function shall select an active ISIM.

If no preference was given and there is more than one active 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 all active UICC applications), from which the end user chooses the UICC application to be selected; if no dialogue is shown the GBA function shall select the active USIM application, if an active USIM application exists, otherwise any active ISIM application.

If no preference was given and there is only one active UICC application, then the GBA function selects this active UICC application;

d. if no "Label" was indicated in step a and if there are no UICC applications active active or if there is no UICC application of the preferred UICC application type active:

- if there is only one UICC application on the UICC, the GBA function selects it, if possible;

- if there is more than one UICC application on the UICC, 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 a preferred UICC application type was indicated and there are UICC applications of this type on the UICC, then the list shown contains only UICC applications of this type, otherwise the list contains all UICC applications on the UICC. If no dialogue is shown the GBA function shall select the "last selected" UICC application of the preferred type (i.e. either the "last selected" USIM or the "last selected" ISIM depending on the given preference), if possible. In case the Ua application indicated "no preference" and both USIM and ISIM are present on the UICC, then the "last selected" USIM is selected.

The procedure in clause 4.4.8.1 shall be followed.

e. if the UICC application type indicated in step a and used in step c and/or d was ISIM, but there was no ISIM to select, then step c and/or d is repeated with UICC application type USIM; otherwise the selection process fails.

NOTE 3: 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 4: 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.

NOTE 5: At any one time, there is at most one UICC application chosen for performing the GBA procedures.

4.4.8.1 UICC application activation procedure in GBA

UICC application activation is defined in TS 31.101 [15].

NOTE: As part of the UICC application (USIM or ISIM) activation procedure, the UICC may require user verification e.g. PIN entry.

If activation of a new UICC application fails then the GBA function shall indicate this to the Ua application.

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

- in the case of GBA_U, the UE and the NAF shall be able to agree which key (i.e, Ks_ext_NAF or Ks_int_NAF or both) is used as the GBA-based shared secret if both keys may be used;

There are two ways to have an agreement between the UE and the NAF which key shall be used Ks_(ext)_NAF or Ks_int_NAF or both:

a) In a generic case, where the protocol used over reference point Ua can be used for different applications (e.g., HTTPS), the protocol should be able to indicate which key should be used.

b) In a specific case, where the protocol is application specific (e.g., MIKEY in MBMS), the agreement can be based on implicit knowledge.

- 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. If a lifetime for the Ks_(ext/int)_NAF (or further adapted key material) is available in the NAF, due to a Ua application specification having its own lifetime value or due to NAF having it's own policy for the adapted key material, then if this lifetime is different from the Ks lifetime received from the BSF, then the NAF shall always select the minimum value for the lifetime out of these two.

- 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. If a lifetime for the Ks_(ext/int)_NAF (or further adapted key material) is available in the NAF, due to a Ua application specification having its own lifetime value or due to NAF having it's own policy for the adapted key material, then if this lifetime is different from the Ks lifetime received from the BSF, then the NAF shall always select the minimum value for the lifetime out of these two.

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.

4.4.11 Requirements on GBA keys and parameters handling

When referring to GBA keys, the following keys are intended: Ks and NAF specific keys derived from the Ks. When referring to NAF specific keys, the following keys are intended: Ks_ext/int_NAF (in GBA_U context) and Ks_NAF (in GBA_ME context), and any keys derived from these keys. The notation Ks_(ext/int)_NAF refers to Ks_ext/int_NAF in GBA_U context and Ks_NAF in GBA_ME context. The notation Ks_(ext)_NAF refers to Ks_ext_NAF in GBA_U context, and Ks_NAF in GBA_ME context.

The ME shall delete all GBA keys (i.e., Ks, and NAF specific keys) and the corresponding NAF_IDs, B-TID, Ks_(int/ext)_NAF lifetimes, Ks lifetime, and lifetime (of the keys derived from Ks_(ext)_NAF) when at least one of the conditions below is met:

1 the UICC is removed from the ME when the ME is in power on state;

2 the ME is powered up and the ME discovers that another UICC has been inserted to the ME. For this, 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 up; or

3 the ME is powered up and the ME discovers that no UICC has been inserted to the ME.

NOTE 1: One possible way, how this requirement can be fulfilled by an application in an open platform is, if the keys are deleted at shut-down and at start-up of the application. When the ME operating system detects one of the conditions above, it can shut down the application to force key deletion. The deletion at start-up ensures that keys are also deleted, when an irregular power-down or UICC removal during power down has occurred.

The ME shall delete all GBA keys related to a certain Ks (i.e., Ks itself, and NAF specific keys derived from this specific Ks) and the corresponding NAF_IDs, B-TID, Ks_(ext/int)_NAF lifetimes, Ks lifetime, and lifetime (of the keys derived from Ks_(ext)_NAF) when the key lifetime of this specific Ks expires.

In the case of GBA_ME, the key Ks shall be deleted from the ME when the ME is powered down. The NAF specific keys (i.e. Ks_(ext)_NAF and keys derived therefrom, if any) may be deleted from the ME when the ME is powered down. If the ME does not delete these NAF specific keys at power down then the NAF specific keys (i.e. Ks_(ext)_NAF and keys derived therefrom, if any) together with the NAF_IDs, B-TID, Ks_(ext)_NAF lifetime and lifetimes (of the keys derived from Ks_(ext)_NAF) shall be stored in non-volatile memory.

If the NAF specific keys are stored in non-volatile memory, then when the ME is powered up again, the ME may need to ensure that the same UICC application is selected for the Ua application, in order to allow the re-use of the NAF specific keys (i.e. Ks_(ext)_NAF and keys derived therefrom, if any), cf. clause 4.4.8. For this, the ME shall store also the IMPI in non-volatile memory. If the same UICC application can not be selected for a Ua application at UE power up, then the ME shall delete the NAF specific keys related to that IMPI stored in non-volatile memory.

Whenever a UICC application is terminated (see section 4.4.8) 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 2: In case the key Ks has been deleted, but the same UICC is still present (i.e. none of conditions 1, 2 or 3 is met), the Ua applications can continue using the NAF specific keys (Ks_(ext/int)_NAF) until the Ks lifetime expires.

4.4.12 Requirements on reference point Zh'

This reference point is optional for the BSF to support. 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, since BSF and HLR are located within the same operator's network.

- the BSF shall be able to send an authentication vector request concerning a subscriber;
- the HLR shall be able to send one authentication vector, as described in TS 29.109 [32] at a time to the BSF;
- no other GBA functionality than conveying authentication vectors shall be required on Zh';
- no state information concerning bootstrapping shall be required in the HLR;
- all procedures over reference point Zh' shall be initiated by the BSF;
- the number of different interfaces to HLR should be minimized.

NOTE 2: If support of GBA User Security Settings (GUSS) is desired in combination with HLR or HSS with Zh' reference point support, then this can be achieved, for instance by storing the GUSS information in a BSF database (external and/or external to the node itself), or in any other network database which is deemed as appropriate for a specific deployment. GUSS information is not sent over Zh' reference point.

4.4.13 Requirements on TMPI handling

The BSF shall store a TMPI together with the IMPI, from which it was derived (cf. Annex B.4), until the next bootstrapping procedure is executed using this TMPI.

The BSF may have a local policy for deleting stored (TMPI, IMPI)-pairs before the next bootstrapping procedure is executed using this TMPI, e.g. for storage or performance reasons.

The ME shall store a TMPI together with the IMPI, from which it was derived (cf. Annex B.4), in non-volatile memory.

The ME shall delete all stored (TMPI, IMPI)-pairs when at least one of the conditions below is met:

1. the UICC is removed from the ME when the ME is in power on state; or
2. the ME is powered up and the ME discovers that another UICC has been inserted to the ME. For this, 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 up; or
3. the ME is powered up and the ME discovers that no UICC has been inserted to the ME.

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 may contact the NAF for further instructions (see figure 4.2).

NOTE: The above text implies that a UE may contact either the BSF or the NAF without knowing whether the NAF supports GBA

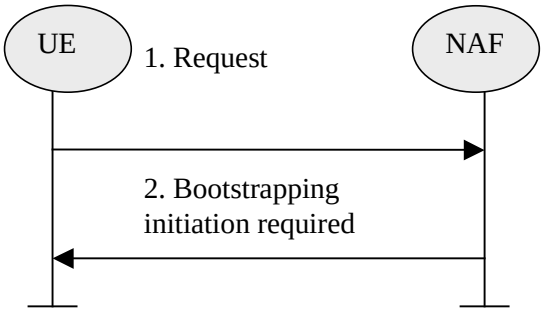


Figure 4.2: Initiation of bootstrapping

1. The UE may start communication over reference point Ua with the NAF with or without any GBA-related parameters.
2. If the NAF requires the use of shared keys obtained by means of the GBA, but the request from UE does not include GBA-related parameters, the NAF replies with a bootstrapping initiation message. The form of this initiation message may depend on the particular reference point Ua.

4.5.2 Bootstrapping procedures

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

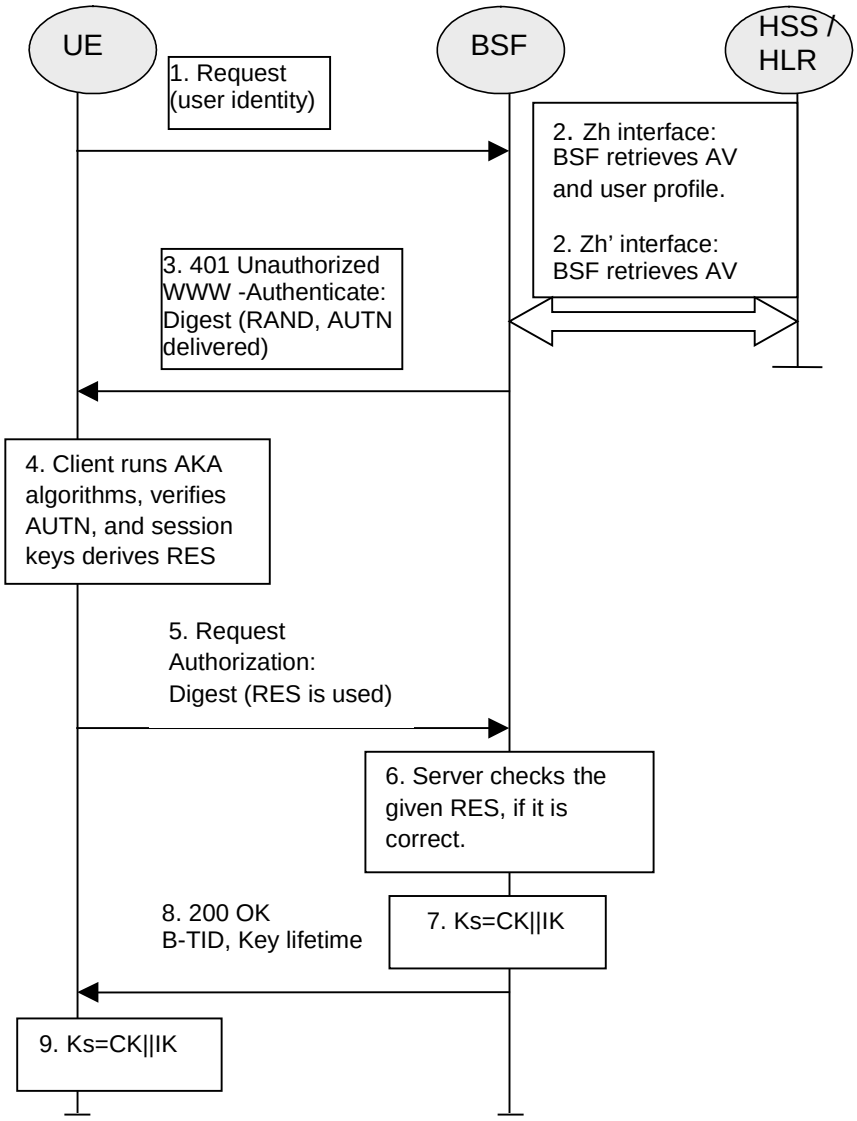


Figure 4.3: The bootstrapping procedure

A UE shall always include the product token "3gpp-gba-tmpi" in the user agent request-header field when communicating over Ub. A BSF shall always include the product token "3gpp-gba-tmpi" in the server response-header field when communicating over Ub.

NOTE 1a: According to the HTTP specification RFC 2616 [33], the product tokens may contain any text. They are ignored when unknown by a UE or a BSF.

1. The UE sends an HTTP request towards the BSF. When a TMPI associated with the IMPI in use is available on the UE, the UE includes this TMPI in the "username" parameter, otherwise the UE includes the IMPI.

2. The BSF recognises from the structure of the "username" parameter (cf. Annex B.4) whether a TMPI or an IMPI was sent. If a TMPI was sent the BSF looks up the corresponding IMPI in its local database. If the BSF does not find an IMPI corresponding to the received TMPI it returns an appropriate error message to the UE. The UE then deletes the TMPI and retries the request using the IMPI.

   The 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.

   In the case that no HSS with Zh reference point is deployed, the BSF retrieves the Authentication Vector over the reference point Zh' from either an HLR or an HSS with Zh' reference point support.

   If the BSF implements the timestamp option and has a local copy of the GUSS for the subscriber that has been fetched from the HSS during a previous bootstrapping procedure, and this GUSS includes a timestamp, the BSF may include the GUSS timestamp in the request message. Upon receiving that timestamp, if the HSS implements the timestamp option, the HSS may compare it with the timestamp of the GUSS stored in the HSS. In this case, if and only if the HSS has done the comparison and the timestamps are equal, then the HSS shall send "GUSS TIMESTAMP EQUAL" indication to the BSF. In any other case, the HSS shall send the GUSS (if available) to the BSF. If the BSF receives "GUSS TIMESTAMP EQUAL" indication, it shall keep the local copy of the GUSS. In any other case, the BSF shall delete the local copy of the GUSS, and store the received GUSS (if sent).

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.

NOTE 3: The password in "AKAv1" HTTP Digest AKA is in binary format.

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 (cf. RFC 3548 [12]) RAND value from step 3, and the BSF server name, i.e. base64encode(RAND)@BSF_servers_domain_name.

NOTE 3a: If the HSS/AuC uses a good random number generator, then the chance of a B-TID collision is practically zero. If such a collision occurs, then the key retrieved by the NAF can have a mismatch with the UE generated NAF key. This will result in a Ua authentication failure which will cause the NAF to once again request the UE to bootstrap which will create a new Ks and a new B-TID.

   If the request included the product token "3gpp-gba-tmpi" in the user agent request-header field the BSF shall compute a new TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

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 is constructed as follows: NAF_Id = FQDN of the NAF || Ua security protocol identifier. The Ua security protocol identifier is specified in Annex H. KDF shall be implemented in the ME.

NOTE 4: If a NAF hosts two or more applications which use the same FQDN and Ua security protocol identifier, they will share the same NAF specific keys. This causes a risk of so called two-time pad which may lead to the situation that the security of these applications is compromised. This can be avoided by running bootstrapping separately to each application or by application specific means, which are however out of the scope of the current specification.

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 [28] 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 TLS Extensions as specified in Annex E of TS 33.310 [19] 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 or until the deletion conditions are satisfied (see 4.4.11).

NOTE 5: The following case can occur. The UE contacts the NAF1 and generates keys for NAF1. Then the UE contacts NAF2 and generates NAF2 keys. Then NAF1 requests then keys from the BSF, but the old key keys could have been overwritten due to NAF2 having initiated a new GBA run. The UE initiates a new GBA-run (B-TID2) after handling NAF1 (B-TID1) and starting the request to the NAF1 over Ua. One possible reason is that B-TID1 lifetime was about to expire. It is very likely that the GBA-run takes much more time (HSS involvement) then the Zn/Ua request such that the B-TID1 request at the BSF should arrive in most cases earlier at the BSF. So this out-of-order case should be very rare. This error situation will be signalled back to the UE, such that the most recent B-TID2 will also be used for NAF1. This out-of order case is self-correcting, since if the B-TID1 is unknown in the BSF, then the Ua request will fail and the UE can send a new request using B-TID2.

If the response included the product token "3gpp-gba-tmpi" in the server response-header field the UE shall compute the TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

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 Ks for the selected UICC application is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause 4.5.2;

     - if no key Ks for the selected UICC application 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

If it is not desired by the UE to use the same Ks for the selected UICC application 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.

- 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 4.5.2, in order to obtain a new key Ks.

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (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 1: 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 4.5.2, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 2: 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.

- the key management procedures for GBA related keys in the ME (i.e. Ks and Ks_NAF keys) are described in section 4.4.11.

- 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;

  According to the procedures defined in clauses 4.5.2 and 4.5.3, 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.;

   - The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

   NOTE 3: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

   - With the key material request, the NAF shall supply a NAF-Id (which includes the NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

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. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to the NAF. 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 4: 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 5: 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.

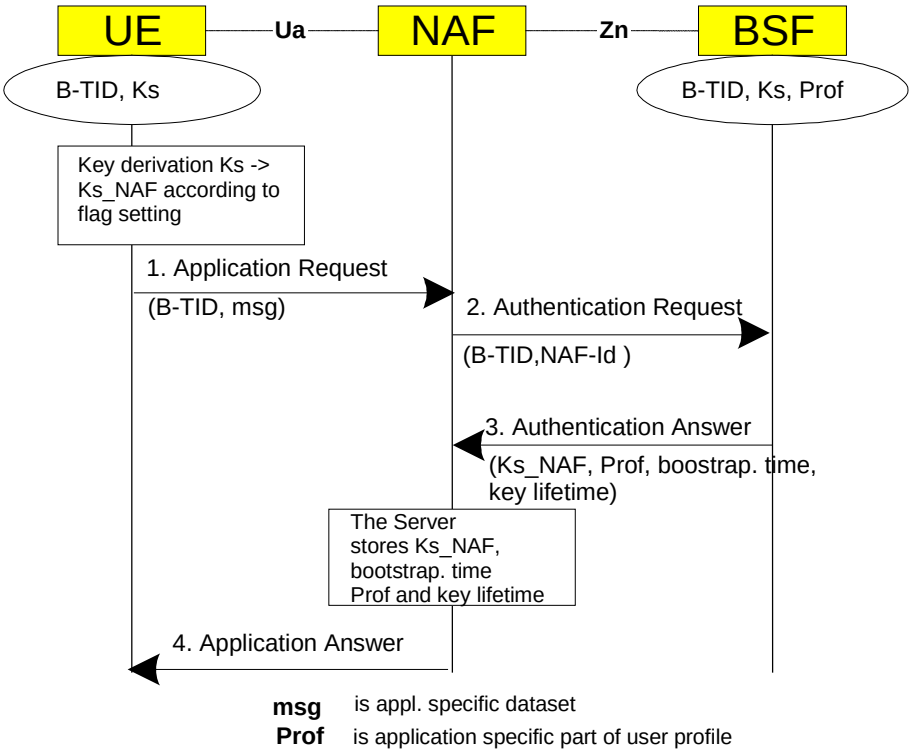


Figure 4.4: The bootstrapping usage procedure

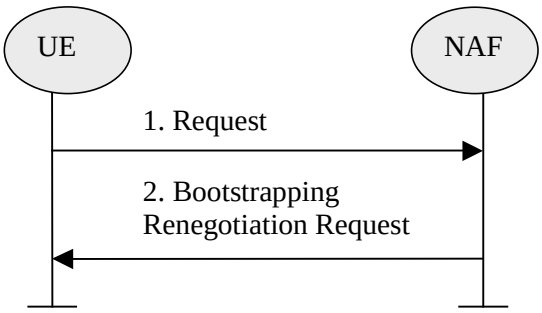


Figure 4.5: Bootstrapping renegotiation request

4.5.4 Procedure related to service discovery

The UE 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 TS 23.003 [11].

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 4.4 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 the present 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.

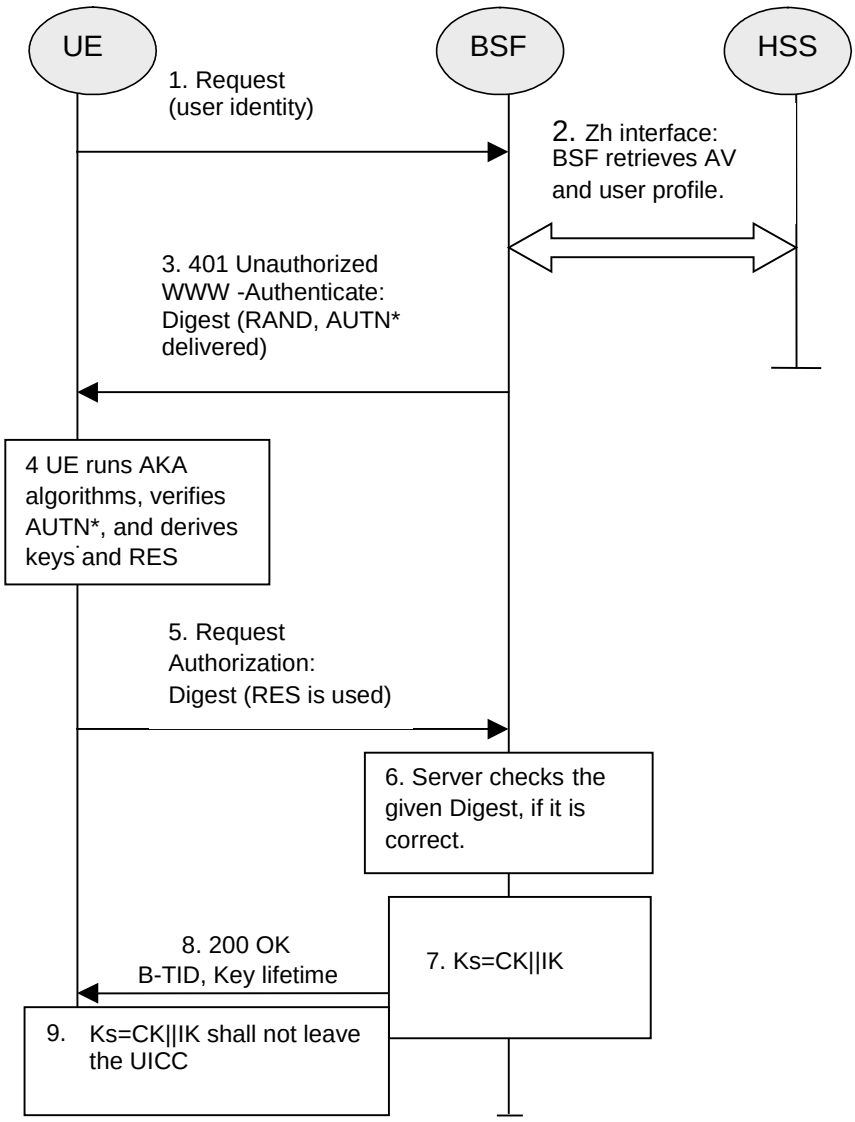


Figure 5.1: The bootstrapping procedure with UICC-based enhancements

A UE shall always include the product token "3gpp-gba-tmpi" in the user agent request-header field when communicating over Ub. A BSF shall always include the product token "3gpp-gba-tmpi" in the server response-header field when communicating over Ub.

NOTE a: According to the HTTP specification RFC 2616 [33], the product tokens may contain any text. They are ignored when unknown by a UE or a BSF.

1. The ME sends an HTTP request towards the BSF. When a TMPI associated with the IMPI in use is available on the UE, the UE includes this TMPI in the "username" parameter, otherwise the UE includes the IMPI.

2. The BSF recognises from the structure of the "username" parameter (cf. Annex B.4) whether a TMPI or an IMPI was sent. If a TMPI was sent the BSF looks up the corresponding IMPI in its local database. If the BSF does not

find an IMPI corresponding to the received TMPI it returns an appropriate error message to the UE. The UE then deletes the TMPI and retries the request using the IMPI.

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.

If the BSF implements the timestamp option and has a local copy of the GUSS for the subscriber that has been fetched from the HSS during a previous bootstrapping procedure, and this GUSS includes a timestamp, the BSF may include the GUSS timestamp in the request message. Upon receiving that timestamp, if the HSS implements the timestamp option, the HSS may compare it with the timestamp of the GUSS stored in the HSS. In this case, if and only if the HSS has done the comparison and the timestamps are equal, then the HSS shall send "GUSS TIMESTAMP EQUAL" indication to the BSF. In any other case, the HSS shall send the GUSS (if available) to the BSF. If the BSF receives "GUSS TIMESTAMP EQUAL" indication, it shall keep the local copy of the GUSS. In any other case, the BSF shall delete the local copy of the GUSS, and store the received GUSS (if sent).

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⊕ Trunc(SHA-1(IK))

NOTE 1: Trunc denotes that from the 160 bit output of SHA-1 [23], 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 ⊕ AK || AMF || 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 IK and MAC (by performing MAC= MAC* ⊕ Trunc(SHA-1(IK))). Then the UICC checks AUTN(i.e. SQN ⊕ AK || AMF || 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.

NOTE 3: The password in "AKAv1" HTTP Digest AKA is in binary format.

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. base64encode(RAND)@BSF_servers_domain_name.

NOTE 3a: If the HSS/AuC uses a good random number generator, then the chance of a B-TID collision is practically zero. If such a collision occurs, then the key retrieved by the NAF can have a mismatch with the UE generated NAF key. This will result in a Ua authentication failure which will cause the NAF to once again request the UE to bootstrap which will create a new Ks and a new B-TID.

If the request included the product token "3gpp-gba-tmpi" in the user agent request-header field the BSF shall compute a new TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

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 = KDF(Ks, "gba-me", RAND, IMPI, NAF_Id), and Ks_int_NAF is computed in the UICC as Ks_int_NAF = 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 is constructed as follows: NAF_Id = FQDN of the NAF || Ua security protocol identifier. The Ua security protocol identifier is 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 4: If a NAF hosts two or more applications which use the same FQDN and Ua security protocol identifier, they will share the same NAF specific keys. This causes a risk of so called two-time pad which may lead to the situation that the security of these applications is compromised. This can be avoided by running bootstrapping separately to each application or by application specific means, which are however out of the scope of the current specification.

To allow consistent key derivation based on NAF name in UE and BSF, at least one of the prerequisites which are specified in clause 4.5.2 shall be met.

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 or until the deletion conditions are satisfied (see 4.4.11).

If the response included the product token "3gpp-gba-tmpi" in the server response-header field the UE shall compute the TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

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. A key selection indication, which key (i.e. Ks_int_NAF or Ks_ext_NAF) the NAF shall use in the Ua reference point may be present in the application specific USS as defined in stage 3 specification. If the indication exists, the NAF shall use the indicated key. If the Ks_int_NAF key was indicated in the USS, the UE attempts to use Ks_ext_NAF key, the NAF shall terminate the communication with the UE.

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.

1. UE starts communication over reference point Ua with the NAF using the keys Ks_ext_NAF or Ks_int_NAF, or both, as required:

   - 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;

  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 2: 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 3: 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 can reply to the first request sent by a UE by sending a key update request to the UE.

- 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

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (cf. clause 4.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (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 4 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.

- the key management procedures for GBA related keys in the ME (i.e. Ks_ext_NAF keys) are described in section 4.4.11.

  - all GBA related keys in the UICC do not need to be deleted when the ME is powered down.

NOTE 5: 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.

  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 6: 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;

- The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

NOTE 7: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

- With the keys request over the Zn reference point, the NAF shall supply a NAF-Id (which includes NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

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. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to

the NAF.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 8: 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 9: 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.

- If the NAF requested an application-specific USS from the BSF and the USS was returned the NAF, the NAF shall check whether this USS contains an key selection indication. If the key selection indication is present, the NAF shall use only the indicated key. If a different key was used over Ua, then the protocol used over reference point Ua shall be terminated.

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.

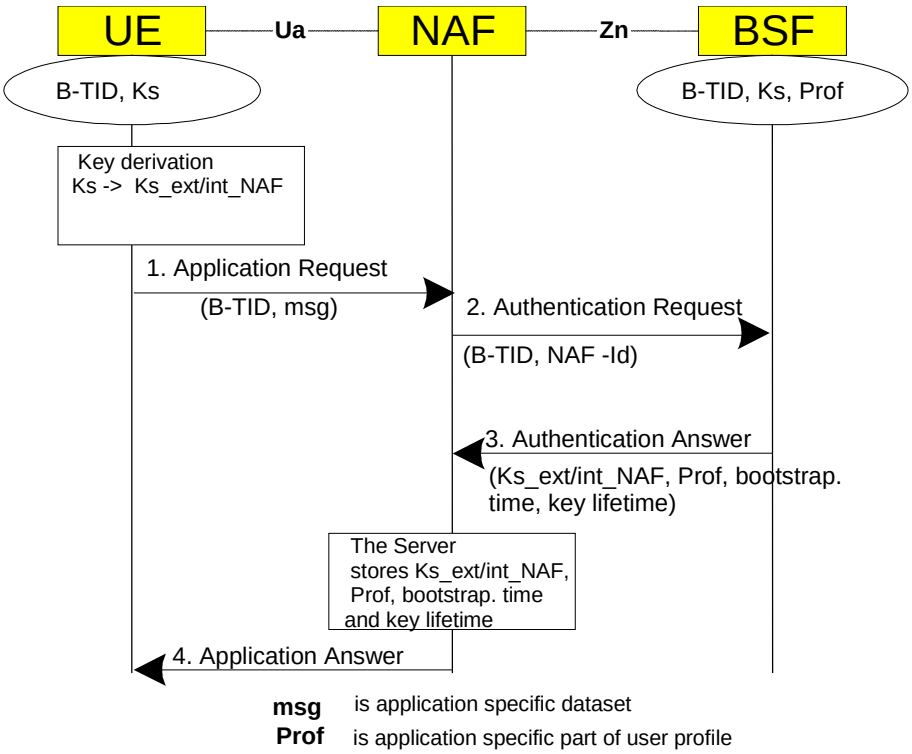


Figure 5.3: The bootstrapping usage procedure with UICC-based enhancements

5.3.4 Procedure related to service discovery

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

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. This KDF is also used to derive the TMPI. 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 measured in octets shall be encoded into a two octet-long string:
   a) express the number of octets in input parameter Pi as a number k in the range [0, 65535].
   b) Li is then a 16-bit long encoding of the number k, encoded as described in clause B.2.1.
2. String S shall be constructed from n+1 input parameters as follows:

   S = FC || P0 || L0 || P1 || L1 || P2 || L2 || P3 || L3 ||... || Pn || Ln

   where

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

   P0 ... Pn are the n+1 input parameter encodings, and

   L0 ... Ln are the two-octet representations of the length of the corresponding input parameter encodings P0.. Pn.

   In this specification the following restriction applies to P0: P0 is a static ASCII-encoded string.

   This restriction is not part of the KDF definition and does not apply to the KDF when used by other 3GPP specifications unless explicitly stated so in those specifications.
3. The final output, i.e. the derived key is equal to the KDF computed on the string S using the key, denoted Key. The present document defines the following KDF:

   derived key = HMAC-SHA-256 (Key , S)

   as specified in [22] and [23].

B.2.1 Input parameter encoding

B.2.1.1 General

This clause specifies how encodings of different data types is to be done. Encoding rules for further data types may be added in future releases if needed.

B.2.1.2 Character string encoding

A character string shall be encoded to an octet string according to UTF-8 encoding rules as specified in IETF RFC 3629 [24] and apply Normalization Form KC (NFKC) as specified in [37].

B.2.1.3 Non-negative integer encoding

A non-negative integer value j, which is input to the KDF, shall be encoded into the octet string Pi as follows:

1. Encode j in base 2.
2. Let n be number of bits as defined for this input parameter in 3GPP specifications, or, if undefined elsewhere, let n be the least multiple of 8 that is greater or equal to the number of bits of the base 2 encoding of j
3. Create an octet string such that the least significant bit of the octet string shall be equal to the least significant bit of j written in base 2, the second to least significant bit of the octet string shall be equal to the second to least significant bit of j written in base 2 and so on (according to clause 3.4 of this specification). If the number of bits in j written in base 2 is less than n, the remaining most significant bits in the octet string shall be set to zero; if the number of bits in j written in base 2 is equal to n, there is no need for such zero padding.

The encoding Li of the number k of octets in Pi follows the above rule.

EXAMPLE1: If Pi is Uplink NAS COUNT then the number k of octets in Pi is 4, according to TS 33.401, Annex A, and n = 32. Assume that Uplink NAS COUNT has the integer value j = 259. Then the base 2 encoding of 259 is 100000011, and Pi consists of the octet string 0000000 0000000 00000001 00000011, or 0x00 0x00 0x01 0x03 in hex representation, and Li consists of the octet string 0000000 00000100, or 0x00 0x04 in hex representation.

EXAMPLE2: If the length of Pi is undefined elsewhere, and the integer value j = 259 is to be encoded into the parameter Pi then the base 2 encoding of 259 is 100000011, n and hence the length of parameter Pi in bits is 16, the number k of octets in Pi is 2, and Pi consists of the octet string 00000001 00000011, or 0x01 0x03 in hex representation, and Li consists of the octet string 0000000 00000010, or 0x00 0x02 in hex representation.

B.2.2 FC value allocations

FC values allocated for this specification shall be in range of 0x00 – 0x0F.

FC values allocated for TS 33.401 [35] shall be in range of 0x10 – 0x1F.

FC values allocated for TS 33.402 [36] shall be in range of 0x20 – 0x2F.

FC values allocated for TS 33.102 [2] shall be in range of 0x30 – 0x3F.

FC values allocated for TS 33.224 [39] shall be in range 0x40 – 0x48.

FC values in range 0x49 – 0xFF are reserved for future use.

B.3 NAF specific key derivation in GBA and GBA_U

In GBA and GBA_U, 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 that 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 that 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).

B.4 Derivation of TMPI

Derivation of TMPI follows the same procedure as NAF specific key derivation in GBA and GBA_U (see clause B.3). As the TMPI is stored in ME, for GBA_U the procedure for derivation of Ks_ext_NAF is followed.

NOTE: This procedure was chosen to avoid any changes to existing UICCs in case of GBA_U.

The BSF_Id defined in this clause consists of the full DNS name of the BSF as used for B-TID generation (see clause 4.5.2), concatenated with the Ua security protocol identifier for TMPI as specified in Annex H.

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

- FC = 0x01,
- P0 = "gba-me" (i.e. 0x67 0x62 0x61 0x2d 0x6d 0x65) (see clause B.3 for Ks_NAF and Ks_ext_NAF),
- L0 = length of P0 is 6 octets (i.e., 0x00 0x06).
- 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 that 65535),
- P3 = the BSF_Id encoded to an octet string using UTF-8 encoding (see clause B.2.1), and
- L3 = length of P3 is variable (not greater that 65535).

The Key to be used in derivation of TMPI shall be:

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

The TMPI is then computed as TEMP@tmpi.bsf.3gppnetwork.org where TEMP is the base64-encoding [12] of the 24 most significant octets of the output of KDF.

B.5 Derivation of passwd and Ks

Derivation of passwd and Ks for GBA_Digest shall follow the same procedure as NAF specific key derivation in GBA and GBA_U as specified in clause B.3.

The input parameters for the key derivation function to derive passwd and Ks shall be the following:

- FC = 0x01,
- P1 = TLS_MK_Extr,
- L1 = length of TLS_MK_Extr is 48 octets (i.e. 0x00 0x30),

In the derivation of passwd as specified in clause M.6.3, step 5,

- P0 = "GBA_Digest_RESP"
  (i.e. 0x47 0x42 0x41 0x5F 0x44 0x69 0x67 0x65 0x73 0x74 0x5F 0x52 0x45 0x53 0x50), and
- L0 = length of P0 is 15 octets (i.e., 0x00 0x0F).

In the key derivation of Ks as specified in clause M.6.3, step 6,

- P0 = "GBA_Digest_Ks"
  (i.e. 0x47 0x42 0x41 0x5F 0x44 0x69 0x67 0x65 0x73 0x74 0x5F 0x4B 0x73),
- L0 = length of P0 is 13 octets (i.e., 0x00 0x0D),
- P2 = RESP, and
- L2 = length of RESP is variable and depends on the algorithm used in HTTP Digest (e.g., 32 if MD5 is used).

The Key to be used in key derivation function shall be:

- H(A1) as specified in clause M.6.3, step 5.

NOTE: In the present document this function is denoted as:
passwd = KDF (H(A1), "GBA_Digest_RESP", TLS_MK_Extr), and
Ks = KDF (H(A1), " GBA_Digest_Ks", TLS_MK_Extr, RESP).

B.6 NAF specific key derivation in GBA_Digest

In GBA_Digest, the input parameters for the key derivation function to derive Ks_NAF shall be the following:

- FC = 0x01;
- P0 = "gba-digest" (i.e. 0x67 0x62 0x61 0x2d 0x64 0x69 0x67 0x65 0x73 0x74);
- L0 = length of P0 is 10 octets (i.e., 0x00 0x0a);
- P1 = nonce;
- L1 = length of nonce is variable (not greater than 65535);
- P2 = IMPI encoded to an octet string using UTF-8 encoding (see clause B.2.1 of the present document);
- 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 of the present document;
- L3 = length of NAF_ID is variable (not greater that 65535).

The Key to be used in key derivation shall be:

- Ks as specified in clause B.5 of the present document.

NOTE: In clause M.6.3 this function is denoted as:
Ks_NAF = KDF (Ks, "gba-digest", nonce, IMPI, NAF_Id).

Annex C:
(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/Zn' reference points

This Annex applies for the Zn' reference point when using DIAMETER or HTTP, and applies for the Zn reference point if using HTTP.

The TLS profile is specified in TS 33.310 [19], Annex E and shall apply with the addition that TLS 1.0 [28] shall also be supported. For all TLS versions the provisions on ciphersuites given in TS 33.310 [19], Annex E, shall apply. The TLS endpoints shall mutually authenticate using certificates as part of TLS session establishment.

NOTE: It is likely that support of TLS 1.0 will no longer be mandatory in a future 3GPP release.

The TLS certificates shall follow the requirements in clause 6.1 of TS 33.310 [19] for TLS certificates, with the exceptions as given in the following.

The Zn-Proxy certificate, i.e. the client certificate used in TLS handshake, shall contain the subjectAltName extension with 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.

The Zn-Proxy certificate shall contain all the DNS names of NAFs that may send a request for NAF specific shared secret through the Zn-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 Zn-Proxy.

Annex F (informative): Handling of TLS certificates

An authentication framework for TLS is available [19].

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.

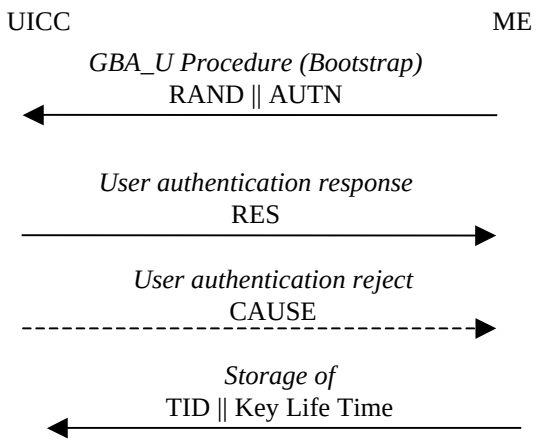


Figure G.1: GBA_U Bootstrap Procedure

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.

In case that the UICC does not have enough storage available for the generated Ks_int_NAF and associated parameters, the UICC shall overwrite an existing Ks_int_NAF entry (Ks_int_NAF and associated parameters). To determine the Ks_int_NAF to overwrite, the UICC shall construct a list of Ks_int_NAF entry numbers by storing in the list first position the entry number of the last used or derived Ks_int_NAF and by shifting down the remaining list elements. The last Ks_int_NAF entry number in this list corresponds to the Ks_int_NAF to overwrite when the UICC runs out of free records.

If an existing Ks_int_NAF entry in use is overwritten, the application Ks_int_NAF shall not be affected (e.g. in case a Ks_int_NAF was put into use as an MBMS MUK key, the MUK key shall continue to be available for the MBMS application).

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 as specified Annex B.2.1.

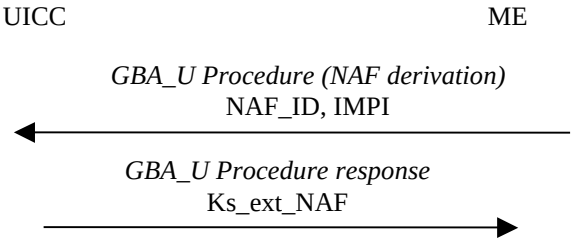


Figure G.2: GBA_U NAF derivation procedure

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 1: 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 2: 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 3: 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 1: 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.
(0x01,0x00,0x00,0x00,0x03)	Ua security protocols used with HTTP-based security procedures for MBMS user services according to TS 26.237 [38].
(0x01,0x00,0x00,0x00,0x04)	Ua security protocols used with SIP-based security procedures for MBMS user services according to TS 26.237 [38].
(0x01,0x00,0x00,0x00,0x05)	Ua security protocols used with Generic Push Layer according to TS 33.224 [39], unless Generic Push Layer is used in the context of another Ua security protocol, which is already covered elsewhere in this Annex.
(0x01,0x00,0x00,0x00,0x06)	Ua security protocol for IMS UE to KMS http based message exchanges according to "IMS media plane security", TS 33.328 [40]
(0x01,0x00,0x00, 0x01,0x00)	Generation of TMPI according to Annex B.4.

NOTE 2: This protocol identifier is not strictly a Ua protocol identifier, but its use in key derivation function is exactly equal.to a Ua protocol identifier.

(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 the IANA TLS Cipher Suite Registry which is referenced in TLS V1.2 [41].

NOTE 3: As an example: RFC 5246 [41] CipherSuite TLS_RSA_WITH_3DES_EDE_CBC_SHA has code = { 0x00,0x0A }, thus the according protocol identifier shall be (0x01,0x00,0x01,0x00,0x0A).

Annex I (normative): 2G GBA

I.0 Introduction

This annex specifies the implementation option to allow the use of SIM cards or SIMs on UICC for GBA. The procedure specified in this annex is called 2G GBA. 2G GBA allows access to applications in a more secure way than would be possible with the use of passwords or with GSM without enhancements. It may be useful for operators who have not yet fully deployed USIMs.

The usage of the term 2G GBA in this specification does not restrict the usage of GBA over only 2G access networks i.e. GSM access. Similarly the use of the term 3G GBA in this specification does not restrict the usage of GBA over only 3G access networks i.e. UMTS. In this specification the term 2G GBA refers to the usage of a SIM card or SIM on UICC, while 3G GBA or GBA on its own, refers and to the usage of a USIM/ISIM on a UICC.

Clauses 4 and 5 of the present document do not apply to this annex unless explicitly stated.

I.1 Reference model

The reference model is the same as described in section 4.1.

I.2 Network elements

I.2.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using the 2G AKA protocol and the TLS protocol, 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 I.5.2.

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

The BSF shall be able to discover from the type of authentication vectors sent by the HSS whether the subscriber is a 2G or a 3G subscriber.

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 I.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.

NOTE 3: If support of GBA User Security Settings (GUSS) for service differentiation is desired in combination with HLR, then this can be achieved by addition of a database to the BSF containing the needed GUSS information.

The BSF shall allow the operator to configure a BSF policy whether to accept 2G subscribers or not for a certain NAF.

I.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 need not be a 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;
- NAF shall have a policy whether to accept 2G subscribers. However, whether the SIM card is allowed to be used with a specific Ua application or not, is dependent on the relevant Ua application. If there is a specific TS for an application using a particular Ua protocol, and unless this TS explicitly prohibits the use of SIM, the operator is allowed to configure a NAF policy whether to accept 2G subscribers or not for this Ua application.

NOTE: Without additional measures, GBA does not guarantee the freshness of the key, Ks(_int/ext)_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(_int/ext)_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.

I.2.2a Zn-Proxy

The text from section 4.2.2a applies also here.

I.2.3 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS.

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 ore 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 from its local database.

NOTE 2: One possibility to revoke temporarily an application specific USS from the GUSS is that the HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber. The GUSS in the BSF is not changed by this operation and only updated when the existing bootstrapping session times out, or is overwritten by a new bootstrapping session during which the new modified GUSS is fetched from HSS along with the AV.

- GUSS shall be able to contain parameters intended for the BSF usage:
  - subscriber specific key lifetime;
  - optionally the timestamp indicating the time when the GUSS has been last modified by the HSS.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or 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.
  - Information on UICC type and on key choice are not required for 2G subscribers. 2G GBA is regarded as ME-based.

I.2.4 UE

The required functionalities from the UE are:

- the support of HTTP Digest AKA protocol;
- the support of TLS;
- the capability to use a SIM in bootstrapping;
- the capability for a Ua application on the ME to indicate to the GBA Function on the ME whether a SIM is allowed for use in bootstrapping (see clause I.4.8);
- the capability to derive new key material to be used with the protocol over Ua interface from Kc, RAND, SRES and Ks-input;
- support of NAF-specific application protocol (For an example see TS 33.221 [5]).

A 2G GBA-aware ME shall support both 3G GBA_U, as specified in clause 5.2 and 3G GBA_ME procedures, as specified in clause 4.5.

I.2.5 SLF

The text from section 4.2.5 applies also here.

I.2.6 HLR

The requirement on the HLR is the same as in clause 4.3.6.

I.3 Bootstrapping architecture and reference points

I.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 2G AKA infrastructure.

I.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 the protocol over reference point Ub.

I.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 reference point Zh is an intra-operator domain interface. The interface to the 2G Authentication Centre is HSS-internal, and it need not be standardised as part of this architecture.

I.3.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous protocol 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.

I.3.5 Reference point Dz

The text from section 4.3.5 applies also here.

I.3.6 Reference point Zh'

The optional reference point Zh' used between the BSF and the HLR allows the BSF to fetch the required authentication information. The reference point Zh' is an intra-operator domain interface.

I.4 Requirements and principles for bootstrapping

I.4.0 General requirements

The following requirements and principles are applicable to bootstrapping procedure:

- the 2G GBA 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.
- Existing SIM cards or SIMs on UICCs and their specifications shall not be impacted.
- If USIM or ISIM are available they shall be used as specified in sections 4 and 5, and 2G GBA shall not be used.
- 2G GBA shall not impact the USIM / ISIM based GBA as specified in sections 4 and 5.
- 2G GBA shall not reduce security for USIM / ISIM users.
- 2G GBA shall minimise the changes to the USIM / ISIM based GBA specified in section 4.
- 2G GBA shall provide measures to mitigate known vulnerabilities of GSM.

I.4.1 Access Independence

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

I.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 GSM authentication (also called 2G AKA) protocol. BSF authentication shall in addition be based on TLS with server certificates.

I.4.3 Roaming

The text from section 4.4.3 applies also here.

I.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 the methods in I.4.2;
- 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.

I.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 since BSF and HSS are located within the same operator's network.

- the BSF shall be able to send bootstrapping information request concerning a subscriber;

- optionally the BSF may have the capability able to send the timestamp of subscriber's GBA user security settings to the HSS (timestamp option);
- the HSS shall be able to send one 2G AKA vector at a time 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. Optionally the HSS may have the capability to indicate to the BSF whether the BSF already has the latest copy of the GUSS based on the GUSS timestamp (timestamp option);

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.

I.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 DIAMETER based Zn reference point shall be secured according to NDS/IP [13];
- If the BSF and the NAF are located in different operators' networks, the DIAMETER based Zn' reference point between the Zn-Proxy and the BSF shall be secured using TLS as specified in Annex E of the present document;
- An HTTP based Zn/Zn' reference point shall be secured using TLS as specified in Annex E of the present document;
- 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 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: Privacy issues need be considered when determining which user identifier is sent to the NAF. If service continuity is desired, then the BSF can be configured to send the IMPI (but then there is no user anonymity). If the BSF does not send the IMPI or IMPU / pseudonym in the USS, then the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user identifier. This can cause that the NAF cannot provide a continuous service, since a user identity is needed in the NAF to ensure that the NAF is able to update keys for a Ua session when the UE has bootstrapped and contacts the NAF with a new B-TID. If user privacy is desired, the NAF can requests a USS and the BSF is configured to send a user pseudonym in the USS, but not the IMPI.

- 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 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).

- The BSF shall send information to the NAF that the subscriber is a 2G subscriber. If no such information is sent the NAF shall assume that the subscriber is a 3G subscriber.

NOTE 6: This requirement enables a NAF to accept 2G subscribers according to its local policy. The second sentence ensures backward compatibility with the procedures specified in section 4 and 5 of this specification. Note also that inclusion of information on the type of subscription in the GUSS would not suffice to satisfy this requirement as a GUSS need not be present for every subscriber.

- The BSF may determine according to its local policy that the NAF shall not serve 2G subscribers. If this is the case, the BSF does not send keys to the NAF.

NOTE 7: This requirement allows an operator controlling the BSF to determine which applications shall use 3G security only. This requirement is also necessary for NAFs, which are not capable to evaluate the information about the subscription type sent by the BSF, e.g. pre-release 7 NAFs.

- NAF shall be able to indicate to BSF the protocol identifier of Ua security protocol it requires the key material by sending NAF-Id to BSF (cf. Annex H).

I.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.

I.4.8 Requirements on selection of UICC application and SIM card

If a UICC is present in the UE, containing a USIM or an ISIM, then a USIM or ISIM shall be used as specified in section 4.4.8. Otherwise a SIM shall be used.

If no UICC, but a SIM card is present in the UE, the SIM card shall be used. The IMPI is obtained from the IMSI as specified in section 4.4.8.

I.4.9 Requirements on reference point Ua

The text from section 4.4.9 applies also here.

I.4.10 Requirements on reference point Dz

The text from section 4.4.10 applies also here.

I.4.11 Requirements on reference point Zh'

The requirements for reference point Zh' are the same as in clause 4.4.12.

I.5 Procedures

This chapter specifies in detail the format of the 2G GBA bootstrapping procedure that is further utilized by various applications. It contains the authentication procedure with BSF, and the key material generation procedure.

I.5.1 Initiation of bootstrapping

The text from clause 4.5.1 of the present document applies also here.

I.5.2 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform a bootstrapping authentication (see figure I.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 I.5.3).

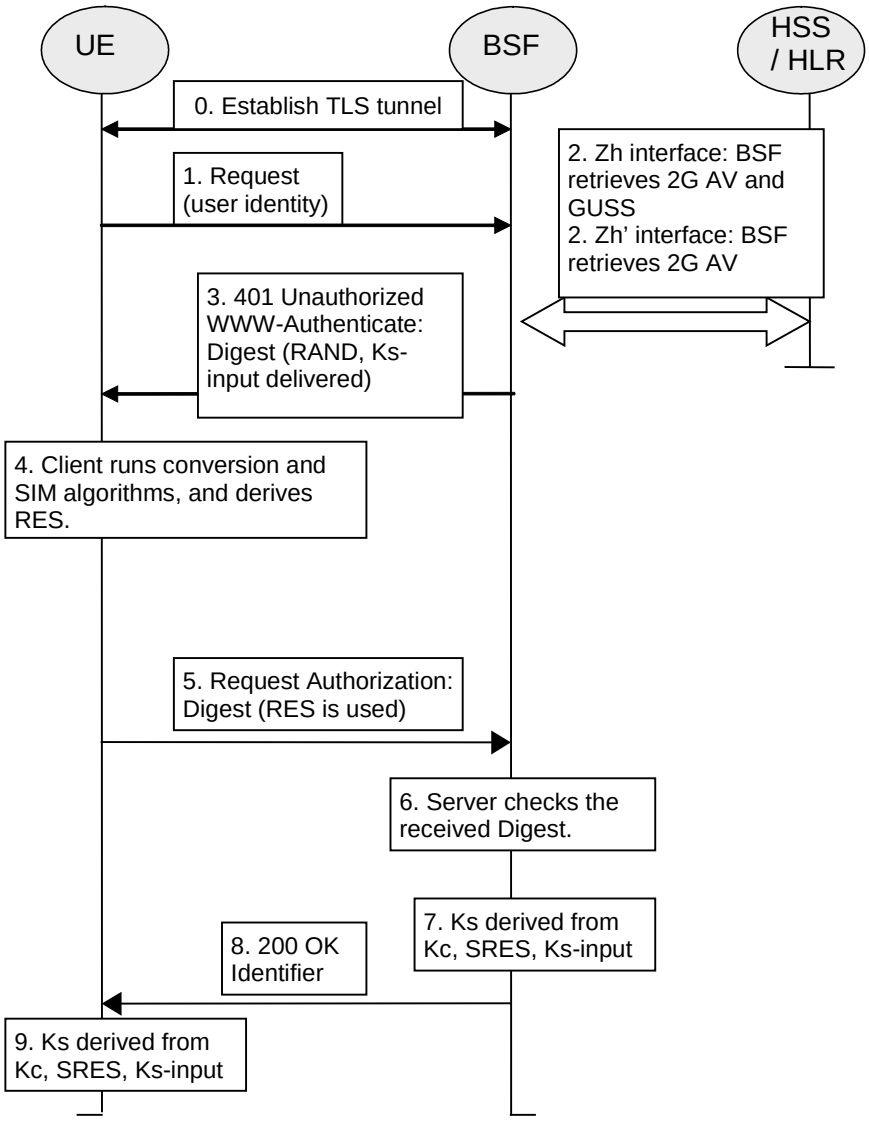


Figure I.3: The bootstrapping procedure

1. The UE sets up a confidentiality-protected TLS tunnel with the BSF. In the set up of the TLS tunnel, the UE shall authenticate the BSF by means of a certificate provided by the BSF. The UE shall check that the "realm" attribute contains the same FQDN of the BSF that was present in the certificate offered by the BSF. All further communication between ME and BSF is sent through this TLS tunnel. The UE now sends an initial HTTPS request.

2. The BSF requests authentication vectors and GUSS from the HSS over Zh. The HSS returns the complete set of GBA user security settings (GUSS) and one 2G authentication vectors (AV = RAND, SRES, Kc) over the Zh reference point. The BSF discovers that the UE is equipped with 2G SIM by looking at the type of authentication vectors.

   If the BSF implements the timestamp option and has a local copy of the GUSS for the subscriber that has been fetched from the HSS during a previous bootstrapping procedure, and this GUSS includes a timestamp, the BSF may include the GUSS timestamp in the request message. Upon receiving that timestamp, if the HSS implements the timestamp option, the HSS may compare it with the timestamp of the GUSS stored in the HSS. In this case, if and only if the HSS has done the comparison and the timestamps are equal, then the HSS shall send "GUSS TIMESTAMP EQUAL" indication to the BSF. In any other case, the HSS shall send the GUSS (if available) to the BSF. If the BSF receives "GUSS TIMESTAMP EQUAL" indication, it shall keep the local copy of the GUSS. In any other case, the BSF shall delete the local copy of the GUSS, and store the received GUSS (if sent).

   In the case that no HSS with Zh reference point support is deployed, the BSF requests the authentication vector from either an HSS with Zh' reference point support or an HLR over the Zh' reference point. The HLR or HSS with Zh' reference point support returns one 2G authentication vectors (AV = RAND, SRES, Kc) over the Zh'

reference point. The BSF discovers that the UE is equipped with 2G SIM by looking at the type of authentication vectors.

The BSF converts one 2G authentication vector (RAND, Kc, SRES) to the parameter RES.

RES = KDF (key, "3gpp-gba-res", SRES), truncated to 128 bits

where key = Kc || Kc || RAND and KDF is the key derivation function specified in Annex B of TS 33.220.

The BSF shall also select a 128-bit random number "Ks-input" and set

server specific data = Ks-input
in the aka-nonce of HTTP Digest AKA, cf. [4].

NOTE 1: "Truncated to 128 bits" means that from the 256 bits output of KDF, the 128 bits numbered as [0] to [127] are used.

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. The BSF shall forward RAND and server specific data in the 401 message to the UE (without RES). This is to demand the UE to authenticate itself.

4. The UE extracts RAND from the message and calculates the corresponding Kc and SRES values. It then calculates the parameter RES from these values as specified in step 2.

5. The UE sends another HTTP request, containing the Digest AKA response (calculated using RES as the password) and a cnonce (cf. [3]), to the BSF.

6. The BSF authenticates the UE by verifying the Digest AKA response. If the authentication fails the BSF shall not re-use the authentication vector in any further communication.

NOTE 3: The password in "AKAv1" HTTP Digest AKA is in binary format.

7. The BSF shall generate key material Ks by computing Ks = KDF (key, Ks-input, "3gpp-gba-ks", SRES).
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. base64encoded(RAND)@BSF_servers_domain_name.

NOTE 3a: If the HSS/AuC uses a good random number generator, then the chance of a B-TID collision is practically zero. If such a collision occurs, then the key retrieved by the NAF can have a mismatch with the UE generated NAF key. This will result in a Ua authentication failure which will cause the NAF to once again request the UE to bootstrap which will create a new Ks and a new B-TID.

8. The BSF shall send a 200 OK message, including a B-TID and an authentication-info header (cf. [3]), 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. The UE shall abort the procedure if the server authentication according to [3] fails. If it is successful the UE shall generate the key material Ks in the same way as the BSF.

10. Both the UE and the BSF shall use the Ks to derive the key material Ks_NAF for use with the procedures specified in clause I.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 is constructed as follows: NAF_Id = FQDN of the NAF || Ua security protocol identifier. The Ua security protocol identifier is specified in Annex H. KDF shall be implemented in the ME.

NOTE 4: If a NAF hosts two or more applications which use the same FQDN and Ua security protocol identifier, they will share the same NAF specific keys. This causes a risk of so called two-time pad which may lead to the situation that the security of these applications is compromised. This can be avoided by running bootstrapping separately to each application or by application specific means, which are however out of the scope of the current specification.

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.

(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 over Ua this requires either multiple-identities certificates for the NAF or the deployment of RFC 3546 [9] over Ua or other protocol means with similar purpose over Ua.

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 or until the deletion conditions are satisfied (see 4.4.11).

I.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 I.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 I.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 for the selected UICC application is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause I.5.2;

  - if no key Ks for the selected UICC application 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;

  If it is not desired by the UE to use the same Ks for the selected UICC application 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;

- 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 I.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 I.5.2, in order to obtain a new key Ks.

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see clause I.5.2). For each protocol used over Ua it shall be specified if only cases (1) and (2) of clause I.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 1: 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 I.3.2, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 2: 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.

- the key management procedures for GBA related keys in the ME (i.e. Ks and Ks_NAF keys) are described in section 4.4.11.

- 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;

  According to the procedures defined in clauses I.5.2 and I.5.3, 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.;

   - The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

   NOTE 3: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

   - With the key material request, the NAF shall supply a NAF-Id (which includes the NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

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 I.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. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to the NAF. In addition, the BSF shall indicate to the NAF that the subscriber is a 2G subscriber. 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 4: 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 5: 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 I.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;

   - If the BSF or the NAF determined, according to their local policies, that the NAF shall not serve 2G subscribers, the NAF shall terminate the protocol over the reference point Ua.

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.

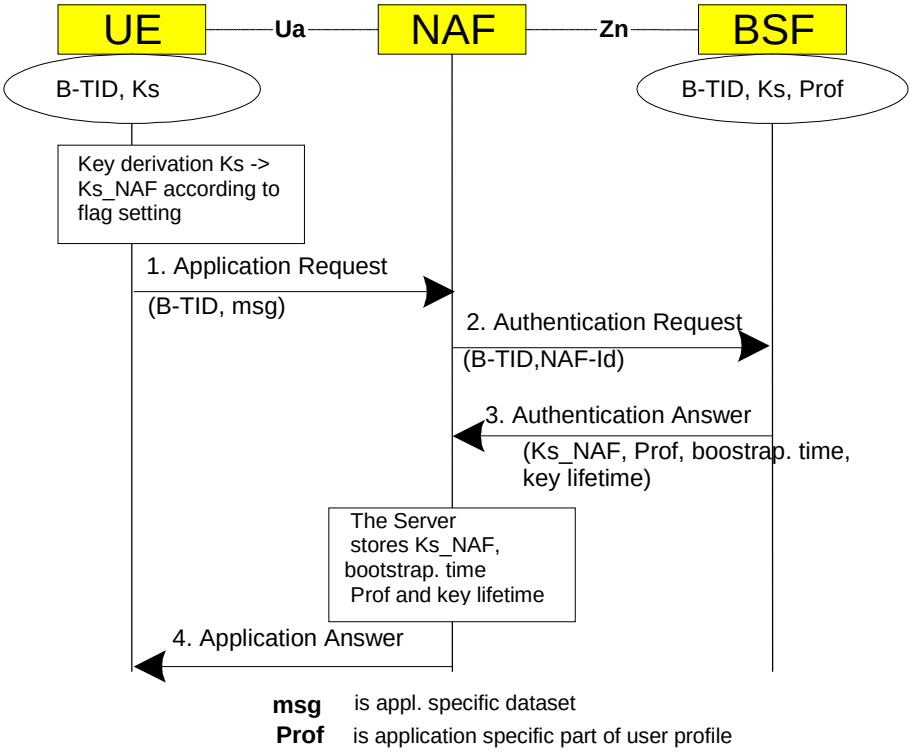


Figure I.4: The bootstrapping usage procedure

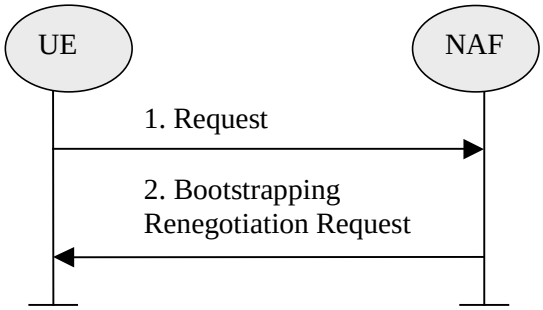


Figure I.5: Bootstrapping renegotiation request

I.5.4 Procedure related to service discovery

The UE shall discover the address of the BSF from the IMSI on the SIM. The same discovery procedure as specified in Section 4.5.4 shall be used.

I.6 TLS Profile

The UE and the BSF shall support TLS according to the TLS profile given in TS 33.310 [19], Annex E.

The certificates shall comply with the requirements for TLS certificates in clause 6.1 of TS 33.310 [19].

Support of certificate revocation and of the related fields in certificates is optional. If supported, the certificate and CRL profiles in clause 6.1 and 6.1a of TS 33.310 [19] should be followed.

NOTE 1: The management of Root Certificates is out of scope of this Technical Specification.

NOTE 2: If no revocation of certificates is deployed, it should be noted, however, that choosing short lifetimes for BSF certificates may considerably reduce the risk, in case BSF certificates may ever be compromised.

I.6.1 void

I.6.2 Authentication of the BSF

The Client shall authenticate the BSF by use of a server certificate. The client shall match the server name as specified in RFC 2818 [18] section 3.1.

The ME shall use a preconfigured list of trusted root certificates for 2G GBA BSF server certificate validation. BSF server certificate validation shall not require manual user interaction.

I.6.3 Authentication of the UE

The BSF shall not request a certificate in a Server Hello Message from the UE. The BSF shall authenticate the UE as specified in clause I.5.2 of this specification.

I.6.4 Set-up of Security parameters

The TLS Handshake Protocol negotiates a session, which is identified by a Session ID. The Client and the BSF shall allow for resuming a session. The lifetime of a Session ID is subject to local policies of the UE and the BSF. A recommended lifetime is five minutes.

NOTE: If the BSF adheres to the recommended lifetime the UE can be certain to be able to resume the TLS session in case of bootstrapping re-negotiation.

Annex J (informative): Usage of USS with local policy enforcement in BSF

This Annex describes how the local policy enforcement in the BSF is used between the NAF and the BSF to control the key delivery to the NAF.

J.1 General

A BSF may have a local policy for zero or more NAFs where the policy for a NAF may state that subscriber's GUSS shall include one or more USSs identified by a GSID. In other words, for a particular NAF the BSF may require that one more USSs shall be present in subscriber's GUSS.

In general, there are two network elements where access control based on some local policy is enforced, i.e. NAF and BSF. Thus two phases with access control based on USSs have to be covered:

1) Access control within NAF for Ua requests: Whether the subscriber is allowed to access the service is decided in the NAF and possibly with the help of USSs. Upon receiving the B-TID from the UE, the NAF fetches the NAF specific shared key (Ks_(ext/int)_NAF) from the BSF, and optionally fetches the USSs, which typically contain NAF specific persistent user identities, and authorization flags. Based on a local policy in the NAF, which may include evaluating the contents of the USS, the NAF decides whether the subscriber is allowed to access the service.

2) Access control within BSF for Zn requests: In certain cases, the operator may wish to implement access control in the BSF. This functionality can be used with any NAF, but the main reason for having this is to implement home operator control in the cases where the NAF is in a visited network.

This Annex describes the access control case within the BSF for Zn requests in more detail.

The following facts should be noted on use of this Annex:

- This access control is completely local to the network of the BSF operator (i.e. home operator of subscriber). This implies that no inter-operator agreement is necessary for implementation of this access control.

- The local policies of the BSF may be based on NAF names and on NAF groups. For the sake of brevity only NAFs are mentioned in the following descriptions.

J.2 Usage scenarios

Four different scenarios can be identified how the local policy enforcement in the BSF will work:

1) A NAF does not use USSs (i.e. it does not request a USS from the BSF), and the BSF does not have a local policy for this NAF.

2) A NAF does not use USSs (i.e. it does not request a USS from the BSF), and the BSF does have a local policy for this NAF.

3) A NAF does use USSs (i.e., it requests one or more USSs from the BSF), and the BSF does not have a local policy for this NAF.

4) A NAF does use USSs (i.e., it request one or more USSs from the BSF), and the BSF does have a local policy for this NAF.

The steps executed in each of these scenarios are described in more detail in the following subclauses.

In all scenarios the NAF has received B-TID from the UE over the Ua reference point before the following steps are executed. The steps describe only the procedures that are related to the local policy enforcement in the BSF with respect to USS existence. Also transfer of other information elements not related to this access control is not mentioned (e.g. key lifetime, private subscriber identity).

J.2.1 Scenario 1: NAF does not use USSs, BSF does not have local policy for NAF

In this scenario, the NAF does not use USSs and the BSF does not have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It does not include any GSIDs in the request.
2. The BSF locates the subscriber information in its local memory using the B-TID.
3. The BSF checks whether a local policy exists for the NAF - in this scenario there is no local policy, i.e. for this particular NAF, the BSF does not require any USSs (identified by GSIDs) to be present in subscriber's GUSS.
4. The BSF derives the NAF specific shared key(s), and sends them to the NAF in the response.
5. The NAF receives the response with the NAF specific shared key(s).

After receiving the NAF specific shared key(s), the NAF may perform access control to the service according to its own policies and continues to communicate with the UE.

J.2.2 Scenario 2: NAF does not use USSs, BSF does have local policy for NAF

In this scenario, the NAF does not use USSs and the BSF does have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It does not include any GSIDs in the request.
2. The BSF locates the subscriber information in its local memory using the B-TID.
3. The BSF checks whether a local policy exists for the NAF - in this scenario there is a local policy for this NAF, i.e. for this particular NAF, the BSF does not require any USSs (identified by GSIDs) to be present in subscriber's GUSS.

   The BSF checks whether all the required USSs identified by GSIDs are present in subscriber's GUSS: If yes, the BSF continues from step 4. If not, the BSF the BSF sends an error message to the NAF.

NOTE: As specified in clause 4.4.6, 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 for particular NAF, rather it is sufficient that the BSF checks the presence of the USSs locally.

4. The BSF derives the NAF specific shared key(s), and sends them to the NAF in the response.
5. The NAF receives the response with the NAF specific shared key(s).

After receiving the NAF specific shared key(s), the NAF may perform access control to the service according to its own policies and continues to communicate with the UE.

If the NAF received the "not authorized" error message, it may indicate this to the UE over Ua reference point. In any case, the GAA based security setup will fail between the UE and the NAF since the NAF did not get the NAF specific shared key(s).

J.2.3 Scenario 3: NAF does use USSs, BSF does not have local policy for NAF

In this scenario, the NAF does use USSs and the BSF does not have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It includes the GSIDs it needs in the request.
2. The BSF locates the subscriber information in its local memory using the B-TID.
3. The BSF checks whether a local policy exists for the NAF - in this scenario there is no local policy, i.e. BSF does not require USSs identified by GSIDs to be present in subscriber's GUSS.

4. The BSF derives the NAF specific shared key(s), and sends them and the USSs identified by the GSIDs to the NAF in the response. If a particular USS is not found in subscriber's GUSS, or the NAF is not authorized to receive a particular USS, these USSs are omitted from the response.

5. The NAF receives the response with the NAF specific shared key(s), and those requested USSs that were available (i.e., found in subscriber's GUSS and allowed by the BSF to be received by the NAF).

After receiving the NAF specific shared key(s) and the available USSs, the NAF may perform access control to the service according to its own policies (e.g. USS required or not, authorization flags required) and continue to communicate with the UE.

J.2.4 Scenario 4: NAF does use USSs, BSF does have local policy for NAF

In this scenario, the NAF does use USSs and the BSF does have a local policy for this NAF.

1. The NAF requests the NAF specific shared key(s) from the BSF. It includes the GSIDs it needs in the request.

2. The BSF locates the subscriber information in its local memory using the B-TID.

3. The BSF checks whether a local policy exists for the NAF - in this scenario there is a local policy for this NAF, i.e., one or more USSs identified by GSIDs shall be present in subscriber's GUSS.

   The BSF checks whether all the required USSs identified by GSIDs are present in subscriber's GUSS: If yes, the BSF continues from step 4. If not, the BSF the BSF sends an error message to the NAF.

4. The BSF derives the NAF specific shared key(s), and sends them and the USSs identified by the GSIDs to the NAF in the response. If a particular USS is not found in subscriber's GUSS, or the NAF is not authorized to receive a particular USS, these USSs are omitted from the response.

5. The NAF receives the response with the NAF specific shared key(s), and those requested USSs that were available (i.e., found in subscriber's GUSS and allowed by the BSF to be received by the NAF).

After receiving the NAF specific shared key(s) and the available USSs, the NAF may perform access control to the service according to its own policies (e.g. USS required or not, authorization flags required) and continue to communicate with the UE.

If the NAF received the "not authorized" error message, it may indicate this to the UE over Ua reference point. In any case, the GAA based security setup will fail between the UE and the NAF since the NAF did not get the NAF specific shared key(s).

Annex K (informative): Interoperator GBA-usage examples

This Annex gives examples how interoperator GBA is set up and operated.

K.1 Example on interoperator GBA setup

Interoperator GBA is set up the following way:

- Each home network operator sets up a BSF, which will enable bootstrapping sessions for its own subscribers.
- Each operator acting as a Serving Network for foreign subscribers in interoperator GBA needs to set up a Zn-Proxy which will forward the authentication requests from its own NAFs to the subscriber's home BSF outside of the VPLMN. The GBA secret is provisioned from the home operator's BSF through the Zn-Proxy to the NAF.

NOTE 1: The security requirements on the Zn' reference point between the Zn-Proxy and the BSF can be found in clause 4.2.2a.

- Each home operator that wants to provide the GBA secrets to foreign NAFs has to authorize these NAFs to request bootstrapping secrets. This is done by using TLS client certificates issued to Zn-Proxies in the serving network by the home network operator.

NOTE 2: The TLS client certificate profile is specified in the normative Annex E.

- An operator that wishes to co-operate in interoperator GBA with another operator shall issue a TLS client certificate to the other operator's Zn-Proxy. Two operators may both act as home operators or as serving operators (i.e., both possess a BSF and a Zn-Proxy), but this Annex also applies to configurations where one operator is always acting as home operator (i.e., hosts the BSF) and the other operator only as serving operator (i.e., the operator hosts only the Zn-Proxy). In the second case, where the serving foreign operator has the Zn-Proxy only, the TLS client certificate is to be handed down in one direction only (see also Annex E on usage of client certificates).

NOTE 3: The handling of TLS certificates is described in TS 33.310 [19]. When two operators sign a roaming agreement, they may also enrol TLS client certificate for each others Zn-Proxies.

NOTE 4: Interoperator GBA is based on bilateral agreements between the two operators. For example, if operator1 has a "GBA agreement" with operator2 and operator1 signs another "GBA agreement" with operator3, this does not mean that operator3 and operator2 have implicitly a "GBA agreement". Operator2 and operator3 shall separately sign a "GBA agreement".

NOTE 5: The home operator may use NAF groups to support local policy checks within its BSF (cf. clause 4.2.1). These may be e.g. one group for NAFs in home network and one group for NAFs in serving networks, or separate groups for each serving network the home operator has "GBA agreements" with. This NAF grouping is under sole responsibility of the home operator and only visible to him. The Zn-Proxies and NAFs in serving networks are not aware of any NAF grouping done in home network.

As described in clause 4.2.2a, a Zn-Proxy may be co-located with a BSF (see Figure K-2). This has the benefit that the NAF has only one logical channel to BSF/Zn-Proxy. Therefore the NAF does not need to make a decision based on the B-TID whether to send the authentication request to the Zn-Proxy or to the BSF. However, this decision can be based on the B-TID as it contains the address of the BSF.

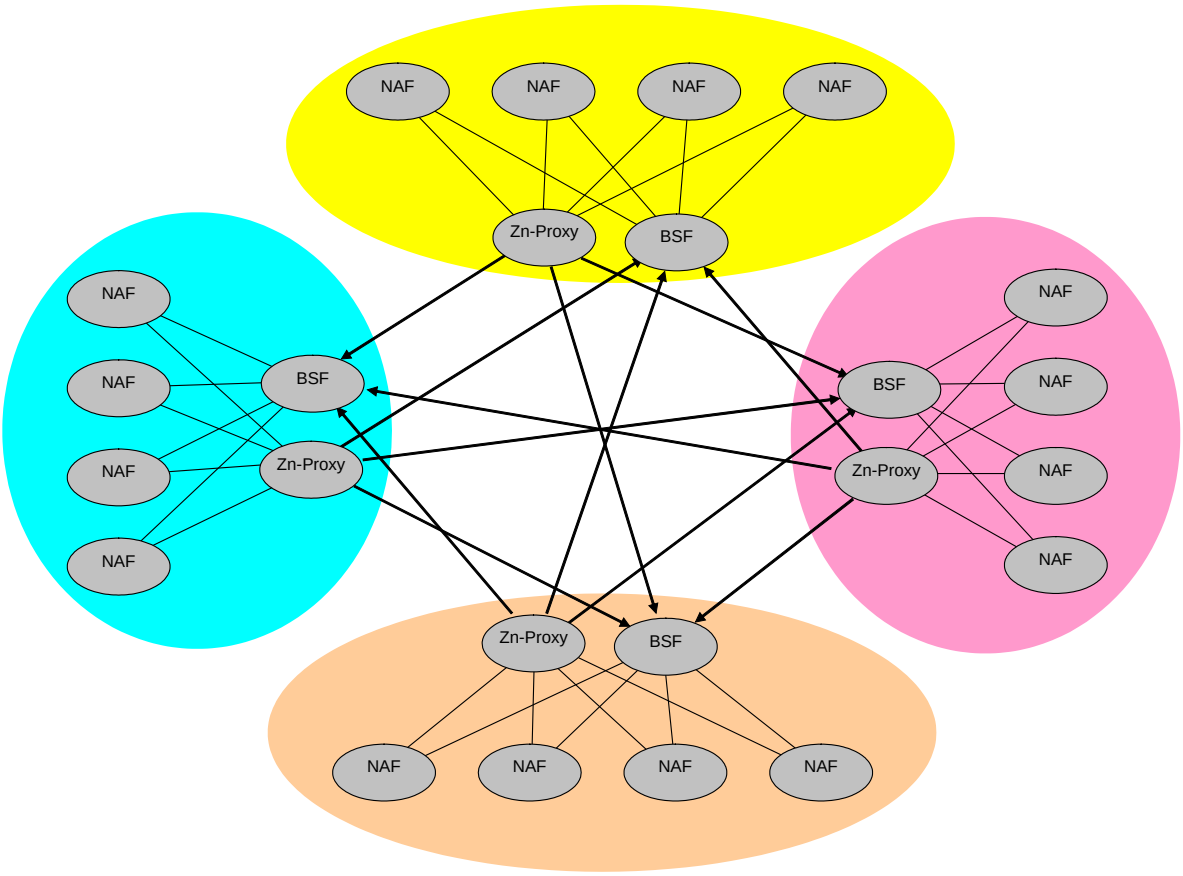


Figure K-1: Interoperator GBA with separate BSF and Zn-Proxy

NOTE 6: The figure K-1 does not show the most general case, where there could be one Zn-proxy per home network in each serving network. It is expected that networks will be optimized and that the existence of one dedicated Zn-proxy for each foreign subscriber home network will be a rare occurrence. The co-location of all Zn-Proxies of one serving network in one location as shown in Figure K-1 is a special case.

NOTE 7: The TLS connections between Zn-Proxy and BSF are "directed", this is indicated in Figure K-1 by the arrowed lines where the arrows point to the server TLS role. The role of the client certificates in these TLS connections is explicitly outlined. Each direction requires a TLS server certificate used at BSF and a TLS client certificate used at Zn-Proxy.

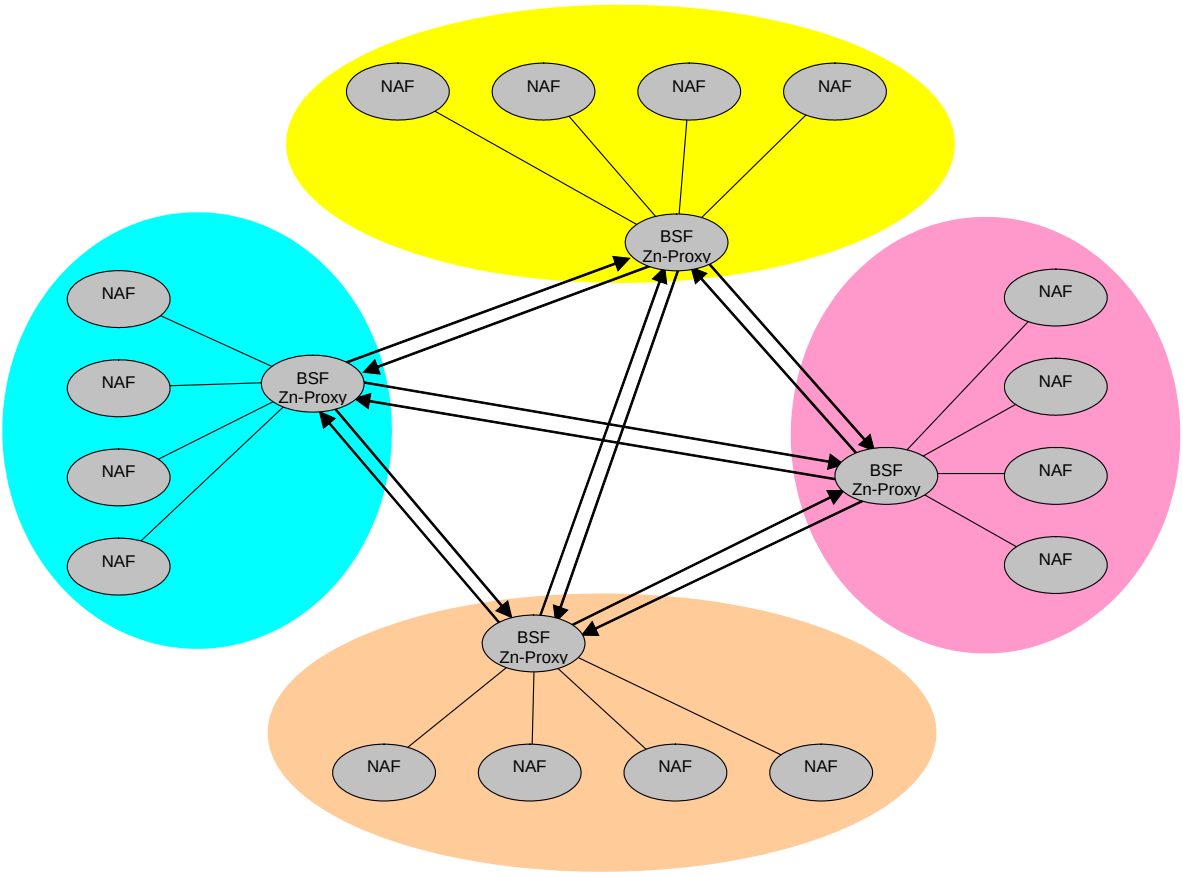


Figure K-2: Interoperator GBA with co-located BSF and Zn-Proxy

NOTE 8: The two distinct TLS connections between Zn-Proxy and BSF are "directed", this is indicated in Figure K-2 by the two lines. Thus the two TLS connection directions may not be intermixed, as the role of the client certificates in these TLS connections is explicitly outlined. Each direction requires a server TLS certificate used at BSF and a client TLS certificate used at Zn-Proxy.

K.2 Example on interoperator GBA operation

Interoperator GBA usage goes as follows:

NOTE 1: This description is based on GBA_ME bootstrapping to simplify the examples, but GBA_U bootstrapping can also be used in interoperator GBA operation.

1. A UE contacts a NAF that does not belong to subscriber's home network. The foreign NAF notifies the UE that 3GPP bootstrapping is required to secure the connection between the UE and the NAF.

2. The UE bootstraps with the home network via the subscriber's BSF. The address of subscriber's home BSF is generated from user's IMSI or IMPI as specified in TS 33.220, clause 4.5.4. The key Ks, and the B-TID are established between the BSF and the UE.

3. The UE derives the NAF specific key Ks_NAF, and uses Ks_NAF and the B-TID on the Ua reference point between the UE and the foreign NAF. At some point during this setup the UE transfers the B-TID to the NAF in the serving network.

4. Upon receiving the B-TID, the foreign NAF has two modes of operations depending on the actual setup of the Zn-Proxy and the BSF in the serving network:

NOTE 2: Any BSF in a network different from the home network of a subscriber and any Zn-Proxy are not visible to the subscriber. To avoid any confusion with the subscribers BSF in the home network, the BSF in a visited network is called foreign BSF in this clause.

a) If the Zn-Proxy and the foreign BSF are separate entities, the foreign NAF shall inspect the B-TID to discover whether the subscriber belongs to its own network, or whether it is a visiting subscriber. In the former case, the request for the Ks_NAF is sent to the BSF, in the latter case, the request is sent to the Zn-Proxy.

b) If the Zn-Proxy and the foreign BSF are a co-located entity, the NAF sends the request for the Ks_NAF to this co-located entity. The NAF does not need to inspect the B-TID.

NOTE 3: Since the B-TID contains the address of subscriber's home BSF, it can be used to discover the home network of the subscriber. A NAF supporting this approach can work with both separated and co-located configurations.

5. Upon receiving the request from the NAF, the Zn-Proxy shall inspect the following:

   b) Validate that the NAF is authorized to request the Ks_NAF (i.e., the DNS part of NAF_Id in the message is correct).

   b) Discover the BSF of the subscriber by inspecting the B-TID.

6. The Zn-Proxy will establish or use the existing DIAMETER or HTTP session to subscriber's home BSF. This DIAMETER or HTTP session is secured by TLS, and the Zn-Proxy shall use a client certificate that the BSF trusts.

7. The Zn-Proxy will forward the request to subscriber's home BSF.

8. Subscriber's home BSF shall validate that the DNS part of the NAF_Id in the request also exists in the client certificate of the Zn-Proxy.

9. Subscriber's home BSF locates the bootstrapping information using the B-TID, processes the request (including possible requests for USSs, local policy check, etc.), derive the NAF specific key, and send the response to the Zn-Proxy.

10. The Zn-Proxy will forward the response to the NAF.

11. The NAF continues with the Ua connection establishment with the UE.

Figure K-3 depicts the entities involved in the above procedure.

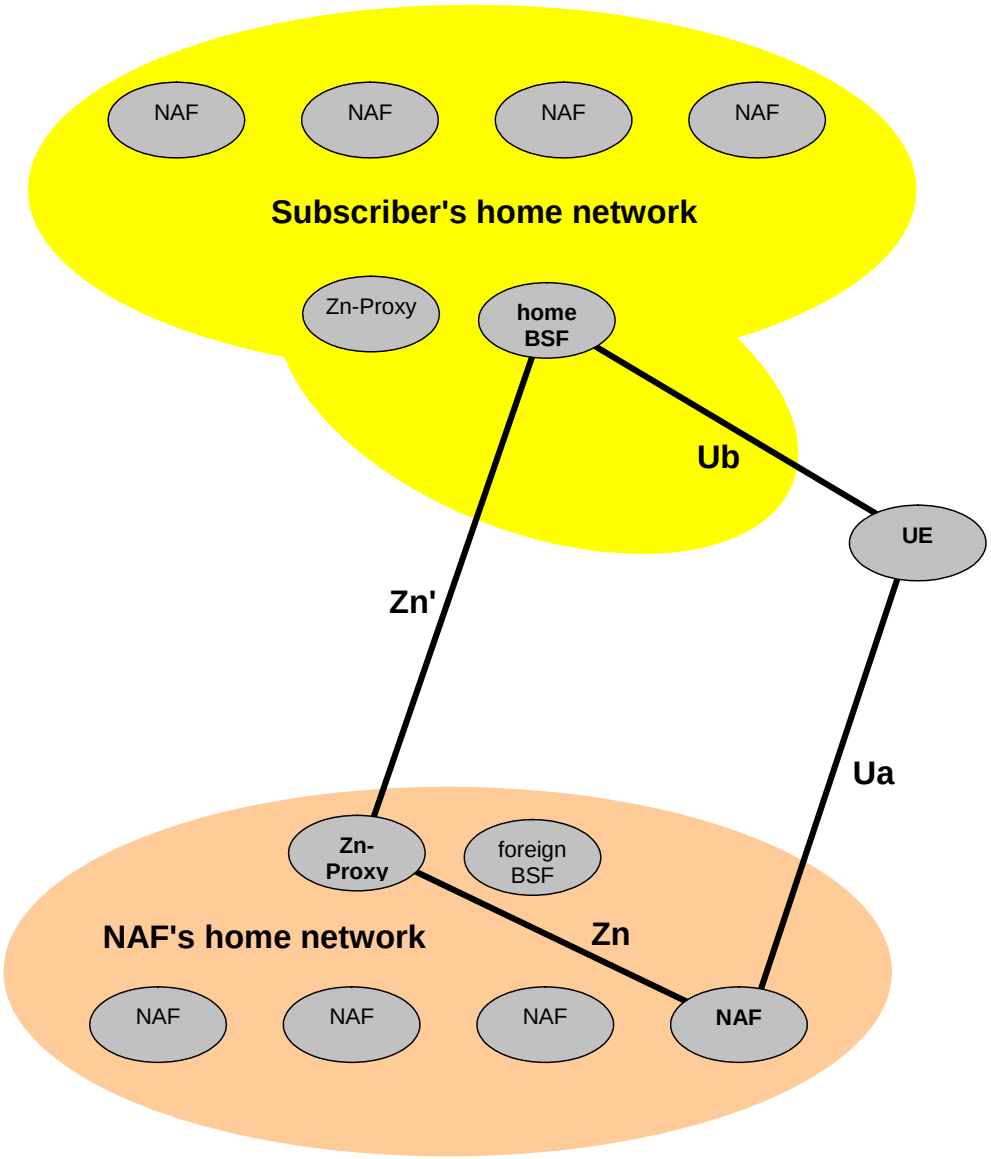


Figure K-3: Interoperator GBA usage

Annex L (informative): Information on how security threats related to known GSM vulnerabilities are addressed by the 2G GBA solution

The 2G GBA solution aims to provide mutual authentication between UE and BSF. This annex examines how the 2G GBA solution mitigates the impersonation of UE or the BSF i.e. security threats related to the known GSM vulnerabilities.

The threats that are originated from the weakness in the usage of the COMP128 algorithm exist independently of the usage of 2G GBA.

L.1 Impersonation of the UE to the BSF during the run of the Ub protocol

This is the main threat to the 2G GBA solution.

1) An attacker (being in the possession of 2G GBA equipment) could try to perform a Man-in-the-middle-attack, impersonating a genuine GSM user to the BSF. In this scenario the attacker would be at the client end of the TLS tunnel to the BSF and send the challenge RAND to the target GSM user, in order to obtain SRES and Kc. However, for the attack to be successful, he would have to find also Kc within the runtime allowed for steps 3 to 5 of the protocol over Ub, as specified in Annex I.5.2. This may be feasible when the terminal of the target GSM user still runs A5/2, but is infeasible for the foreseeable future when one of the other GSM encryption algorithms is used. A5/2 will be removed from networks by the end of 2006, and will not be present in any 2G GBA enabled terminals. A vulnerability caused by A5/2 would only exist in the case where a GSM user has subscribed to 2G GBA feature, but uses his SIM in an old terminal with A5/2 enabled while being targeted by the attacker. But the practical implications of this remaining vulnerability are expected to be limited as a user subscribed to 2G GBA will own a Release 7 terminal (2G GBA will be a Release 7 feature), and the likelihood of him inserting his SIM in an old terminal, and an attacker obtaining this information and exploiting it for a man-in-the-middle attack, may be low in practice. Furthermore, old terminals will gradually disappear.

2) SIM cloning: an attacker being able to find the long-term key Ki of a genuine GSM user is able to fully impersonate him in all contexts, including the 2G-GBA one (if this has been subscribed by the genuine user).. The attacker could do this by exploiting weaknesses of A3/A8 as they were found for COMP128, while in possession of the SIM i.e. the attacker tries to find the long term key K. Even if 2G GBA does not increase the risk of possible A3/A8 breakages, it has to be noted that the COMP128-related issue disappears when more secure A3/A8 algorithms are used. These are available today, cf. "GSM MILENAGE", as specified in TS 55.205 v610. Operators are advised in general to discontinue the use of COMP128

3). Unauthorized access to SIM needs to be countered by platform security methods. The impacts of a compromised SIM/ME or UICC/ME interface on GAA security are similar in 2G GBA and 3G GBA.

L.2 Impersonation of the BSF to the UE during the run of the Ub protocol

To prevent an impersonation attack of the BSF to the UE during the run of the Ub protocol the authentication of the BSF to the UE is improved by protecting the communication with TLS. An attacker succeeds only if he can break both, the certificate-based TLS authentication to the UE and mutual authentication provided by HTTP Digest using a password derived from GSM procedures. One way to break TLS is to compromise the certificate.

L.3 Finding the GBA key Ks during or after the Ub protocol run

For BSF-to-UE authentication and for establishment of the key Ks, the solution relies on both, GSM security and TLS security. The attacker needs to know all the parameters of the GSM triplet, in particular Kc, and additionally break the TLS security, as the attacker also needs to know the Ks-input parameter confidentially transmitted by the BSF over TLS. Breaking GSM security after the Ub protocol run alone does not provide sufficient information to break 2G GBA.

L.4 Bidding down attack

To avoid a bidding down attack (also called downplay attack), the 2G GBA solution requires that a GBA-enabled terminal that supports SIM based 2G GBA must support also USIM/ISIM based 3G GBA as specified in I.2.4. If a USIM/ISIM is available, then the terminal must use the USIM/ISIM based 3G GBA as specified in I.4.8.

Annex M (normative): GBA_Digest

M.1 General

This annex specifies the use of SIP Digest credentials, as defined in TS 33.203 [16], for GBA. The procedure specified in this annex is called GBA_Digest. GBA_Digest allows access to applications in a more secure way than would be possible with the use of password-based HTTP Digest as specified in RFC 2617 [3] without enhancements. It may be useful for environments where a UICC, or a SIM card, is not available to subscribers. The use of GBA_ Digest is restricted to such environments.

Clauses 4 and 5 of the present document do not apply to this Annex unless explicitly stated.

NOTE: The use of the term 'UE' in this Annex is in line with the use of the term 'UE' in TS 33.203 [16], Annex N (on SIP Digest), but differs from that in other 3GPP specifications in that it assumes that a UICC is not available to subscribers in the UE.

M.2 Reference model

The reference model is the same as described in clause 4.1, with the exception that the reference point Zh' is not needed here.

M.3 Network elements

M.3.1 Bootstrapping server function (BSF)

A generic Bootstrapping Server Function (BSF) and the UE shall mutually authenticate using a combination of the HTTP Digest protocol and the TLS protocol, 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 M.6.3.

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

The BSF shall discover from the request received from the UE over the Ub interface whether the UE intends to run GBA_Digest. The BSF shall then request a SIP Digest authentication vector from the HSS or abort the Ub run with a suitable failure message, according to its local policy.

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 I.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.

The BSF shall allow the operator to configure a BSF policy whether to accept subscribers using SIP Digest credentials or not for a certain NAF.

M.3.2 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 need not be a previous security association between the UE and the NAF;
- NAF shall locate and communicate securely with the subscriber's BSF;
- NAF shall 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;
- NAF shall have a policy whether to accept subscribers using SIP Digest credentials. However, whether GBA_Digest is allowed to be used with a specific Ua application or not, is dependent on the relevant Ua application. If there is a specific TS for an application using a particular Ua protocol, and unless this TS explicitly prohibits the use of GBA_Digest, the NAF may allow usage of SIP Digest credentials for this application,
- the NAF shall be able to indicate to the UE that the SIP Digest-based GBA bootstrapping security association is acceptable.

NOTE: Without additional measures, GBA, as defined throughout the present document, does not guarantee the freshness of the key, Ks_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_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.

M.3.3 Zn-Proxy

The text from clause 4.2.2a applies also here.

M.3.4 HSS

The set of all user security settings (USSs), i.e. GUSS, is stored in the HSS.

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 ore 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 from its local database.

NOTE 2: One possibility to revoke temporarily an application specific USS from the GUSS is that the HSS may temporarily remove an application-specific USS from the GUSS if the service is temporarily revoked from the subscriber. The GUSS in the BSF is not changed by this operation and only updated when the existing bootstrapping session times out, or is overwritten by a new bootstrapping session during which the new modified GUSS is fetched from HSS along with the AV.

- GUSS shall be able to contain parameters intended for the BSF usage:
  - subscriber specific key lifetime;
  - optionally the timestamp indicating the time when the GUSS has been last modified by the HSS.

NOTE 3: These parameters are optional and if they are missing from subscriber's GUSS or 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.
- Information on UICC type and on key choice are not required for subscribers using SIP Digest credentials. GBA_Digest is regarded as ME-based.

M.3.5 UE

The required functionalities from the UE are:

- the support of HTTP Digest protocol according to RFC 2617 [3] with the additional profiling specified in this Annex;
- the support of TLS;
- the capability to use SIP Digest credentials in bootstrapping;

- the capability for a Ua application on the terminal to indicate to the GBA Function on the terminal whether SIP Digest credentials are allowed for use in bootstrapping;
- the capability to derive new key material to be used with the protocol over the Ua interface as defined in clause M.6.3;
- support of at least one Ua application protocol (For an example see TS 33.221 [5]);
- the capability to send an indication to the BSF over the Ub interface that the UE intends to run GBA_Digest.

M.3.6 SLF

The text from clause 4.2.5 applies also here.

M.4 Bootstrapping architecture and reference points

M.4.1 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 SIP Digest credentials.

M.4.2 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 the protocol over reference point Ub.

M.4.3 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 reference point Zh is an intra-operator domain interface.

M.4.4 Reference point Zn

The reference point Zn is used by the NAF to fetch the key material agreed during a previous protocol 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.

M.4.5 Reference point Dz

The text from clause 4.3.5 applies also here.

M.5 Requirements and principles for bootstrapping

M.5.1 General Requirements

The following requirements and principles are applicable to bootstrapping procedure:

- the GBA_Digest 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;
- If USIM, ISIM, or SIM are available and the BSF supports AKA-based GBA the UE shall not use GBA_Digest. Instead, the UE shall use the procedures as specified in clauses 4 and 5, and Annex I;
- GBA_Digest shall not impact the procedures for AKA-based GBA as specified in clauses 4 and 5, and Annex I;
- GBA_Digest shall not reduce security for users of AKA-based GBA;
- GBA_Digest shall be closely modelled after AKA-based GBA specified in clauses 4 and 5, and Annex I;
- GBA_Digest shall provide measures to mitigate known vulnerabilities of the re-use of SIP Digest credentials.

M.5.2 Access independence

The bootstrapping procedure for GBA_Digest is, in principle, access independent as it only requires IP connectivity from the UE. However, in order to ensure that GBA_ Digest is not used over access networks defined in 3GPP specifications operators may introduce some access dependence in their network configurations, e.g. by assigning different ports on the BSF to different access networks.

M.5.3 Authentication methods

Authentication between the UE and the BSF shall not be possible without a valid IMS subscription. Authentication shall be based on a combination of the HTTP Digest protocol using SIP Digest credentials and the TLS protocol, as defined in clause M.6.3. TLS shall be used with server certificates, but the TLS server shall not request client certificates.

M.5.4 Roaming

The requirements on roaming are:

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

M.5.5 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 authenticate each other based on the methods specified in clasue M.5.3;
- the BSF shall send a bootstrapping transaction identifier to the UE;
- the UE and the BSF shall establish shared keys;

- the BSF shall 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.

M.5.6 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 since BSF and HSS are located within the same operator's network.

- the BSF shall send a bootstrapping information request concerning a subscriber;
- optionally the BSF may have the capability to send the timestamp of subscriber's GBA user security settings to the HSS (timestamp option);
- the HSS shall send one SIP Digest authentication vector at a time to the BSF;
- the HSS shall send the complete set of subscriber's GBA user security settings needed for security purposes to the BSF. Optionally the HSS may have the capability to indicate to the BSF whether the BSF already has the latest copy of the GUSS based on the GUSS timestamp (timestamp option);

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 the HSS should be minimized.

M.5.7 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 DIAMETER based Zn reference point shall be secured according to NDS/IP in TS 33.210 [13];
- If the BSF and the NAF are located in different operators' networks, the DIAMETER based Zn' reference point between the Zn-Proxy and the BSF shall be secured using TLS as specified in Annex E;
- An HTTP based Zn/Zn' reference point shall be secured using TLS as specified in Annex E;
- 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 send a key material request to the BSF, containing NAF's public hostname used by the UE's corresponding request. The BSF shall verify that a NAF is authorized to use this hostname, i.e. the FQDN used by UE when it contacts the NAF;
- The NAF shall indicate to the BSF for each Zn run whether it is willing to accept Ks_NAF based on GBA_Digest;
- The BSF shall 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 indicate to the BSF the single application or several applications it requires USSs for;

NOTE 1: If some application needs only a subset of an application-specific USS the NAF selects this subset from the complete set of USS sent from BSF.

- The BSF shall 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 2: Privacy issues need be considered when determining which user identifier is sent to the NAF. If service continuity is desired, then the BSF can be configured to send the IMPI (but then there is no user anonymity). If the BSF does not send the IMPI or IMPU / pseudonym in the USS, then the user remains anonymous towards the NAF; or more precisely, the B-TID functions as a temporary user identifier. This can cause that the NAF cannot provide a continuous service, since a user identity is needed in the NAF to ensure that the NAF is able to update keys for a Ua session when the UE has bootstrapped and contacts the NAF with a new B-TID. If user privacy is desired, the NAF can requests a USS and the BSF is configured to send a user pseudonym in the USS, but not the IMPI.

- 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 configure the BSF in such a way that no USS is required for the requesting NAF;

- The BSF shall 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 3: This does not preclude a NAF to refresh the key before the expiry time according to the NAF's local policy.

NOTE 4: 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).

- If the NAF indicated its willingness to accept Ks_NAF based on GBA_Digest in the Zn request and the B-TID sent by the NAF points to a Ks generated by GBA_Digest the BSF shall send information to the NAF that the subscriber is a subscriber who used SIP Digest credentials. If the B-TID points to a Ks established by another GBA method the BSF shall respond according to that method. Otherwise, the BSF shall not send key material to the NAF.

NOTE 5: This requirement enables a NAF to accept subscribers using SIP Digest credentials according to its local policy. The second sentence ensures backward compatibility with the procedures specified in clauses 4 and 5 and Annex I. Note also that inclusion of information on the GBA variant in the GUSS is not possible as one subscriber may have both AKA and SIP Digest credentials, leading to a depencence on the credentials actually used during the last Ub run.
A NAF that can understand a 'GBA_Digest' indication received from the BSF on Zn can understand which GBA variant was used on Ub to derive the Ks_NAF key and, hence, can always make its own judgment whether to accept the Ks_NAF based on its local policy. However, there is no technical reason why the NAF would not accept a Ks_NAF that was derived using an AKA-based GBA variant because such a Ks_NAF is stronger than a key that was derived using GBA_digest and there is no difference in using it for the NAF.

- The BSF may determine according to its local policy that the NAF shall not serve subscribers using SIP Digest credentials. If this is the case, the BSF shall not send keys generated by GBA_Digest to the NAF.

NOTE 6: This requirement allows an operator controlling the BSF to determine which applications shall use AKA-based GBA only.

- The NAF shall indicate to the BSF the protocol identifier of Ua security protocol for which it requires the key material by sending NAF-Id to BSF (cf. Annex H).

M.5.8 Requirements on Bootstrapping Transaction Identifier

The text from clause 4.4.7 applies also here.

M.5.9 Requirements on reference point Ua

The text from clause 4.4.9 applies also here.

M.5.10 Requirements on reference point Dz

The text from clause 4.4.10 applies also here.

M.5.11 Requirements on GBA keys and parameters handling

- The terminal shall delete all GBA keys related to a certain Ks (i.e., Ks itself, and NAF specific keys derived from this specific Ks) and the corresponding NAF_IDs, B-TID, , Ks lifetime, and, if applicable, Ks_NAF lifetimes and lifetimes of the keys derived from a Ks_NAF, when the key lifetime of this specific Ks expires.

M.6 Procedures

M.6.1 General

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

M.6.2 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 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 may contact the NAF for further instructions (see figure M.1).

NOTE: The above text implies that a UE may contact either the BSF or the NAF without knowing whether the NAF supports GBA.

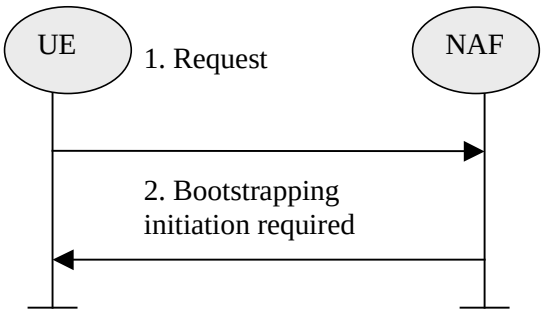


Figure M.1: Initiation of bootstrapping

1. The UE may start communication over reference point Ua with the NAF with or without any GBA-related parameters.

2. If the NAF requires the use of shared keys obtained by means of the GBA, but the request from UE does not include GBA-related parameters, the NAF replies with a bootstrapping initiation message. If the use of GBA_Digest is acceptable to the NAF the NAF shall indicate it in this message. The form of this initiation message may depend on the particular reference point Ua.

M.6.3 Bootstrapping procedures

When a UE wants to interact with a NAF, and it knows that the bootstrapping procedure is needed, it shall first perform such a procedure. Otherwise, the UE shall perform a bootstrapping procedure only when it has received a bootstrapping initiation required message or a bootstrapping negotiation indication from the NAF, or when the lifetime of the key in UE has expired (cf. clause M.6.4).

The bootstrapping procedure using SIP Digest credentials is run over the Ub interface (extended for the purposes of GBA_Digest) as described below:

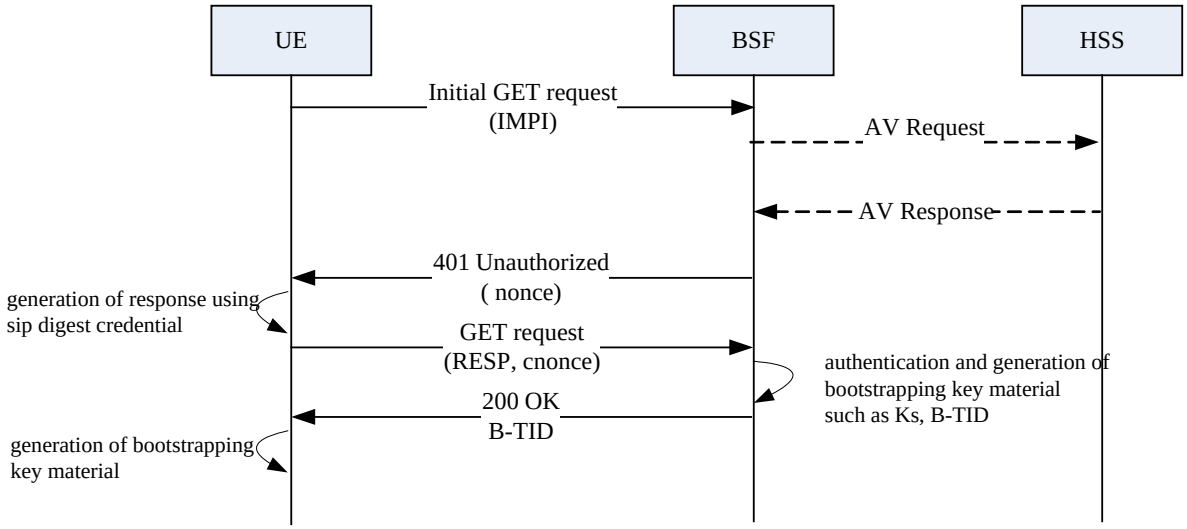


Figure M.2 GBA_Digest bootstrapping procedure

NOTE 1: Figure M.2 only shows an example flow for visualization and not all details are included.

A UE shall always include the product token "3gpp-gba-tmpi" in the user agent request-header field when sending HTTP messages over Ub. A BSF shall always include the product token "3gpp-gba-tmpi" in the server response-header field when sending HTTP messages over Ub.

NOTE 1a: According to the HTTP specification RFC 2616 [33], the product tokens may contain any text. They are ignored when unknown by a UE or a BSF.

Step 0:

The UE and the BSF shall establish a TLS tunnel with server authentication using a server certificate. The use of TLS message integrity is mandatory, while the use of TLS encryption is optional. All further messages between the BSF and UE shall be sent through this tunnel.

NOTE 2: TLS encryption can be useful for protecting the user identity privacy when the TMPI mechanism defined in the present document is not used.

Step 1:

In this HTTP request message from the UE to the BSF, the UE shall include an Authorization header containing a user identity in the "username" parameter and a token indicating the use of GBA_Digest. When a TMPI associated with the IMPI in use is available on the UE, this user identity shall be this TMPI, otherwise it shall be the IMPI. The realm in the Authorization header shall be the realm as defined for SIP Digest in TS 33.203 [16].

Step 2:

The BSF recognises from the structure of the "username" parameter (cf. Annex B.4) whether a TMPI or an IMPI was sent. If a TMPI was sent the BSF shall look up the corresponding IMPI in its local database. If the BSF does not find an IMPI corresponding to the received TMPI it shall return an appropriate error message to the UE. The UE shall then delete the TMPI and retry the request using the IMPI.The BSF shall request a SIP Digest Authentication Vector (SD-AV) from the HSS. The SD-AV is defined in TS 33.203 [16], Annex N. The username field in the Multimedia Auth Request shall contain the IMPI.

Step 3:

The HSS shall retrieve the SD-AV corresponding to the IMPI and send it to the BSF in a Multimedia Auth Answer. The handling of GUSS between BSF and HSS shall be as described in clause 4.5.2, step 2.

The qop value shall be set to "auth-int ".

NOTE 3: The additional protection afforded by qop set to "auth-int" may seem unnecessary considering the fact that the messages exchanged between UE and BSF are protected by a TLS tunnel. However, the use of "auth-int" is consistent with the other modes of GBA (GBA_ME, GBA_U and 2G GBA) and also provides a second layer of integrity protection in case the TLS server authentication is ever compromised (e.g. due to replacement of insecurely stored root certificates on the UE or a Certification Authority being compromised).

Step 4:

In the HTTP 401 Unauthorized response from the BSF to the UE, the BSF shall include a WWW-Authenticate header with parameters as specified in RFC 2617 [3].

The parameters realm, qop, and algorithm were provided in the SD-AV in step 3 and the nonce=base64encode (16 byte random value) is generated according to RFC 3548 [12] by the BSF.

Step 5:

When responding to a challenge from the BSF, the UE shall generate a cnonce randomly, and calculate the response RESP. The RESP shall be put into the Authorization header and sent back to the BSF in the GET request.

RESP shall be computed as a Digest-response according to RFC 2617 [3] (HTTP Digest) from the most recent GBA_Digest challenge and a password 'passwd' that is generated as follows:

passwd = KDF (H(A1), "GBA_Digest_RESP", TLS_MK_Extr)

where H(A1) is the hash of the following three parameters: the user name and password used by the user in IMS for SIP Digest according to TS 33.203 [16], Annex N, and the realm, cf. also RFC 2617 [3]. "GBA_Digest_RESP" is a character string. TLS_MK_Extr is extracted from the TLS master key according to RFC5705 [44] with the optional context value being omitted, the label set to "EXPORTER_GBA_Digest", and the length set equal to the length of the TLS master secret (48 bytes). KDF is the key derivation function as specified in clause B.2.

NOTE 4: A cautionary note on notation: According to RFC 2617 [3], the computation of RESP from the password 'passwd' defined above entails again a parameter called H(A1). This parameter will differ from the value of H(A1) that is input to the above formula because the passwords from which these two H(A1) values are derived differ. But no new notation is deemed necessary here as the notation H(A1), when H(A1) is derived from 'passwd', is not explicitely used in the text of the present document.

Step 6:

Upon receiving a GET request carrying the authentication response RESP, the BSF shall check that the expected RESP (calculated by the BSF in the same way as by the UE in step 5) matches the received RESP. If the check is successful then the user has been authenticated.

The BSF shall then derive Ks as follows, cf. also clause B.x:

Ks = KDF (H(A1), "GBA_Digest_Ks", TLS_MK_Extr, RESP)

where H(A1), RESP, and TLS_MK_Extr are defined as in step 5, and "GBA_Digest_Ks" is a character string.

The BSF shall generate the bootstrapping transaction identifier (B-TID) for the IMPI and store the tuple <B-TID, IMPI, Ks, nonce>. The B-TID shall be constructed in the format of a NAI by taking the nonce from step 4, and the BSF server name, i.e. nonce@BSF_server_domain_name.

NOTE 5: The B-TID construction above is almost identical to the one used in clause 4. The difference is that in clause 4 the username part is constructed from the (base64 encoded) RAND value.

The BSF shall compute a new TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

NOTE 6: The formulations in the preceding paragraph, and the corresponding paragraph below relating to the computation of the TMPI in the UE, differ from the ones in clause 4.5.2 as GBA_Digest-aware UEs and BSFs always include the product tokens as described at the start of this clause. So, the condition in clause 4.5.2 is not needed.

The BSF shall send a 200 OK response to the UE to indicate the success of the authentication.

In this message from the BSF to the UE, the BSF shall include the bootstrapping transaction identifier (B-TID) and the key lifetime.

An Authentication-Info header according to RFC 2617 [3] shall be included into the 200 OK response.

The UE shall abort the procedure if the server authentication according to RFC 2617 [3] fails. Otherwise, the UE shall derive Ks in the same way as the BSF did above.

The UE shall compute the TMPI as specified in Annex B.4 and store it together with the IMPI, overwriting a previous TMPI related to this IMPI, if any.

After successful bootstrapping procedure the UE and the BSF shall store the key Ks, the nonce, the B-TID, and an indication of the underlying security quality, i.e. GBA_Digest, for further use, until the key Ks is updated or until the deletion conditions in clause M.5.11 are satisfied. The key Ks shall then be used in the BSF and in the UE to derive NAF specific key(s) Ks_NAF to secure Ua reference points in the following way:

Ks_NAF shall be computed as Ks_NAF = KDF (Ks, "gba-digest", nonce, IMPI, NAF_Id), where KDF is the key derivation function as specified in clause B.2, and the input parameters consist of the user's IMPI, the NAF_Id and 'nonce'. 'nonce' is the nonce that was used for computing the RESP that was input to the derivation of Ks. The NAF_Id shall be constructed as in clause 4.5.2. The "gba-digest" parameter is a static character string.

NOTE 6: The above derivation of Ks_NAF is analagous to the derivation in clause 4.5.2, step 9, and the same KDF can be utilized.

The KDF shall be implemented in the terminal.

M.6.4 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 M.6.2.

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 M.3.

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, and if the use of a Ks derived from an AKA-based GBA variant according to clauses 4.5.3, 5.5.3, or I.5.3, is not possible, the UE proceeds as follows:

  - if the UE knows (through a lack of indication in the Initiation of Bootstrapping procedure or by configuration) that the use of GBA_Digest is not acceptable to the NAF it shall abort the communication with the NAF. Otherwise, a key Ks_NAF shall be derived in the following way:

  - if a key Ks derived from SIP Digest credentials is available in the UE, the UE derives the key Ks_NAF from Ks, as specified in clause M.6.3;

  - if no key Ks derived from SIP Digest credentials is available in the UE, the UE first agrees on a new key Ks derived from SIP Digest credentials with the BSF over the reference point Ub, and then proceeds to derive Ks_NAF;

NOTE 0: A key Ks derived from an AKA-based GBA variant could still be available from a previous GBA bootstrapping run where the UICC was available, and could then still be used.

If it is not desired by the UE to use the same Ks derived from SIP Digest credentials 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;

- 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 M.4. 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 M.6.3, in order to obtain a new key Ks.

  To allow for consistent key derivation in BSF and UE, both have to use the same FQDN for derivation (see clause M.6.3). For each protocol used over Ua it shall be specified if only cases (1) and (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 1: 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 M.5.8, to allow the NAF to retrieve the corresponding keys from the BSF;

NOTE 2: 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 the present document.

- the key management procedures for GBA related keys in the terminal are described in section M.5.11.

- 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;

  According to the procedures defined in clauses M.6.3 and M.6.4, 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 shall request key material corresponding to the B-TID supplied by the UE to the NAF over reference point Ua. The NAF shall indicate to the BSF whether it is willing to accept Ks_NAF based on GBA_Digest;

   - The NAF may also request one or more application-specific USSs for the applications, which the request received over Ua from UE may access;

   NOTE 3: If the NAF requires service continuity, then the NAF can request a USS that contains a user pseudonym that allows service continuity according to BSF policy.

   - With the key material request, the NAF shall supply a NAF_Id (which includes the NAF's FQDN that the UE has used to access this NAF and the Ua security protocol identifier) to the BSF. (This is to allow for consistent

key derivation in the BSF and UE as described above). The BSF shall verify that the NAF is authorized to use that FQDN.

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 M.6.3, 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. For any USSs containing a NAF Group attribute, this attribute shall be removed in the USSs supplied to the NAF. In addition, the BSF shall indicate to the NAF that the subscriber is a subscriber using SIP Digest credentials. 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 4: 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 5: 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 the present document.

   - If the NAF did not indicate that it is willing to accept a Ks_NAF based on GBA_Digest, or if the BSF determines according to its local policy that the NAF shall not serve subscribers using SIP Digest credentials, then the BSF shall not send a Ks_NAF based on GBA_Digest;

   - If the NAF indicated that it is willing to accept a Ks_NAF based on GBA_Digest, but the B-TID refers to a key Ks established by using an AKA-based method, then the BSF shall send a key Ks_NAF derived from this Ks unless this Ks was derived from 2G GBA and the NAF does not accept 2G GBA (cf. NOTE 0);

   - 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 M.5.7). 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;

   - If the NAF determines, according to its local policy, that the NAF shall not serve subscribers using SIP Digest credentials, the NAF shall terminate the protocol over the reference point Ua;

   - The NAF should accept the Zn response even when the GBA_Digest indication is missing (as this means that the key Ks_NAF was derived from a key Ks established by using an AKA-based method, which is stronger), (cf. NOTE 0).

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.

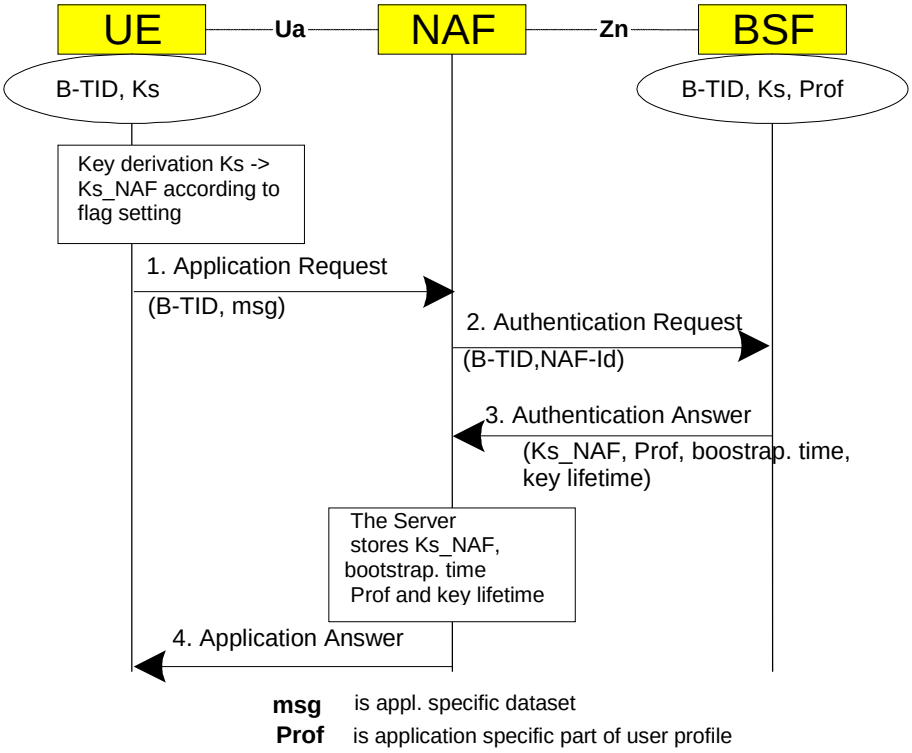


Figure M.3: The bootstrapping usage procedure

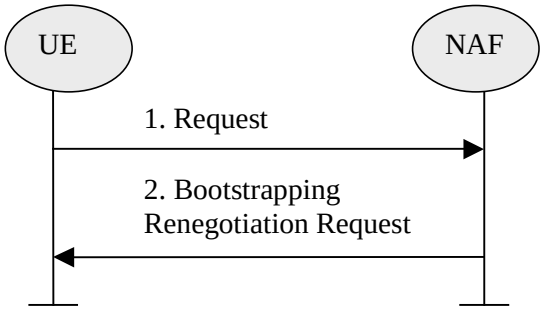


Figure M.4: Bootstrapping renegotiation request

M.6.5 Procedure related to service discovery

The UE shall discover the address of the BSF from the IMPI related to the IMS subscription. When the IMPI was derived from an IMSI as defined in clause 13 of TS 23.003 [11] then the BSF address shall be derived as as specified in clause 16 of TS 23.003 [11] for the case of an IMSI, otherwise the BSF address shall be derived as as specified in clause 16 of TS 23.003 [11] for the case of an IMPI.

NOTE: The reason for this distinction is the NOTE in clause 16 of TS 23.003 [11] warning that BSF addresses of a certain form may be unreachable.

M.7 TLS Profile

M.7.1 General

The UE and the BSF shall support TLS according to the TLS profile given in TS 33.310 [19], Annex E.

The certificates shall comply with the requirements for TLS certificates in clause 6.1 of TS 33.310 [19].

Support of certificate revocation and of the related fields in certificates is optional. If supported, the certificate and CRL profiles in clauses 6.1 and 6.1a of TS 33.310 [19] should be followed.

NOTE 1: The management of Root Certificates is out of scope of the present document.

NOTE 2: If no revocation of certificates is deployed, it should be noted, however, that choosing short lifetimes for BSF certificates may considerably reduce the risk, in case BSF certificates may ever be compromised.

M.7.2 Authentication of the BSF

The Client shall authenticate the BSF by use of a server certificate. If the BSF also supports 2G GBA under the same BSF address (cf. TS 23.003 [11]) it is recommended that this certificate be the same as the one used for BSF authentication in 2G GBA, cf. clause I.6.2. The client shall match the server name as specified in RFC 2818 [18], section 3.1.

NOTE: If the BSF addresses derived for the 2G GBA case and the GBA_Digest case differ, but the BSF is the same, then the operator can issue a BSF certificate with two server names or with a wildcard server name to ensure the client check of the server name works correctly, or the operator can issue two different BSF certificates.

The terminal shall use a preconfigured list of trusted root certificates for GBA_Digest BSF server certificate validation. It is recommended that this list be the same as the one used for 2G GBA BSF server certificate validation, cf. clause I.6.2. BSF server certificate validation shall not require manual user interaction.

M.7.3 Authentication of the UE

The BSF shall not request a certificate in a Server Hello Message from the UE. The BSF shall authenticate the UE as specified in clause M.6.3.

M.7.4 Set-up of Security parameters

The TLS Handshake Protocol negotiates a session, which is identified by a Session ID. The Client and the BSF shall allow for resuming a session. The lifetime of a Session ID is subject to local policies of the UE and the BSF.

Annex N (informative): Change history

Change history									
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**	**WI**
2004-03	SP-23	SP-040175	-	-	D	Presented for approval at TSG SA #23	1.2.1	2.0.0	
2004-03	SP-23	-	-	-	F	Approved and placed under Change Control (Rel-6)	2.0.0	6.0.0	
2004-06	SP-24	SP-040375	001	-	F	Removal of Annex A	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040376	002	-	B	NAF remove the security associations	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040377	003	1	D	Removal of editors notes on Transaction Identifiers	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040378	004	1	B	Introduction of a UICC-based Generic Bootstrapping Architecture	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040379	005	-	D	Editorial corrections to TS 33.220	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040380	006	-	C	Support for NAF in visited network	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040381	007	-	C	Editorial changes and clarifications to TS 33.220	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040382	008	-	F	Multiple key derivation mandatory	6.0.0	6.1.0	SEC1--SC
2004-06	SP-24	SP-040383	009	-	C	NAF's public hostname verification	6.0.0	6.1.0	SEC1--SC
2004-09	SP-25	SP-040619	010	-	C	Detailing of key lifetime	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	011	-	C	Details of USIM/ISIM usage in GAA	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	012	-	C	Generic Ua interface requirements	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	013	-	F	B-TID generation	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	014	-	B	Securing Zn reference point	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	015	-	D	GBA User Security Settings	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	016	-	C	Creation of GBA_U AV in the BSF	6.1.0	6.2.0	SEC1-SC
2004-09	SP-25	SP-040619	017	-	F	Clarification of the definition of a default type of NAF-specific key	6.1.0	6.2.0	SEC1-SC
2004-12	SP-26	SP-040855	018	1	C	BSF discovery using default domain method	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	019	1	C	Local validity condition set by NAF	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	020	3	C	GBA User Security Settings (GUSS) usage in GAA and Introduction of NAF groups	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	021	2	C	Details of USIM/ISIM selection in GAA	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	023	-	B	TLS profile for securing Zn' reference point	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	025	2	F	Optimization of the GBA_U key derivation procedure	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	027	2	F	Requirement on ME capabilities for GBA_U	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	034	1	D	Adding a note about replay protection	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	035	1	C	Complete the MAC modification for GBA_U	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	036	1	F	Removal of unnecessary editor's notes	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	038	1	C	Fetching of one AV only on each Zh run between BSF and HSS	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	039	1	B	Clean up of TS 33.220	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	040	1	F	New key management for ME based GBA keys	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	041	1	C	Key derivation function	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	042	1	D	Re-negotation of keys	6.2.0	6.3.0	SEC1-SC
2004-12	SP-26	SP-040855	043	1	C	No GUSS/USS update procedures in Release-6	6.2.0	6.3.0	GBA-SSC
2004-12	SP-26	SP-040855	044	1	C	Clarify the number of NAF-specific keys stored in the UE per NAF-Id	6.2.0	6.3.0	SEC1-SC
2005-03	SP-27	SP-040139	045	1	F	Key derivation function: character encoding	6.3.0	6.4.0	SEC1-SC
2005-03	SP-27	SP-040139	047	1	D	Bootstrapping timestamp	6.3.0	6.4.0	SEC1-SC
2005-03	SP-27	SP-040139	048	-	F	Storage of B-TID in GBA_U NAF Derivation procedure	6.3.0	6.4.0	SEC1-SC
2005-06	SP-28	SP-050262	050	1	F	Usage of USS for local policy enforcement in BSF	6.4.0	6.5.0	SEC1-SC
2005-06	SP-28	SP-050262	051	1	F	Correcting figure 4.4	6.4.0	6.5.0	SEC1-SC
2005-06	SP-28	SP-050263	052	-	B	GBA User Security Settings (GUSS) transfer optimisation	6.4.0	7.0.0	SEC1-SC
2005-09	SP-29	SP-050553	0054	-	A	Clarification of anonymous access to NAF in GBA	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050554	0056	-	A	Removing IMPI from USS	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050572	0057	-	C	Informative annex on usage of USS for local policy enforcement in BSF	7.0.0	7.2.0	SEC1-SC
2005-09	SP-29	SP-050557	0059	-	A	Removing duplication of text relating to BSF addressing	7.0.0	7.1.0	SEC-SC1
2005-09	SP-29	SP-050555	0061	-	A	Clarification of lifetime of derived keys	7.0.0	7.1.0	SEC1-SC

2005-09	SP-29	SP-050575	0062	-	B	Introduction of key selection mechanism	7.0.0	7.2.0	SEC1-SC
2005-09	SP-29	SP-050556	0064	-	A	Addition of the Dz interface for multiple HSS deployments	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050565	0066	-	A	Removing requirement to send authentication vectors in batches	7.0.0	7.1.0	GBA
2005-09	SP-29	SP-050551	0068	-	A	Clarification concerning input parameter encoding for GBA_U NAF derivation procedure	7.0.0	7.1.0	SEC1-SC
2005-09	SP-29	SP-050577	0069	1	B	Normative annex on 2G GBA	7.0.0	7.1.0	2G GBA
2005-09	SP-29	SP-050552	0071	-	A	Providing Ua-security protocol based key separation	7.0.0	7.1.0	SEC1-SC
2005-10	post SP-29	-	-	-	-	Editorial change to align annexes between Release 6 and Release 7 based on CRs at SP-29	7.1.0	7.1.1	
2005-12	SP-30	SP-050768	0073	-	A	NAF_Id encoding	7.1.1	7.2.0	SEC1-SC
2005-12	SP-30	SP-050775	0074	-	B	Informative annex with examples on interoperator GBA usage	7.1.1	7.2.0	GAA2
2005-12	SP-30	SP-050775	0075	-	F	Clarification of local policy enforcement	7.1.1	7.2.0	GAA2
2005-12	SP-30	SP-050777	0076	-	F	Alignment of 2G GBA with recent CRs	7.1.1	7.2.0	2GGBA
2005-12	SP-30	SP-050777	0077	-	F	Addition of information requested by SA plenary on 2G GBA	7.1.1	7.2.0	2GGBA
2005-12	SP-30	SP-050777	0078	-	F	IMPI obtained from IMSI in 2G GBA	7.1.1	7.2.0	2GGBA
2005-12	SP-30	SP-050775	0079	-	F	Removal of possible interoperability problems	7.1.1	7.2.0	GAA2
2006-03	SP-31	SP-060061	0080	-	B	D-proxy renaming to Zn-Proxy	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-03	SP-31	SP-060061	0082	-	B	Protection of Zn/Zn' reference point for http based protocols	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-03	SP-31	SP-060051	0084	-	A	Restricting the TLS CipherSuites in Annex E and cleanup of references	7.2.0	7.3.0	SEC1-SC
2006-03	SP-31	SP-060061	0085	-	F	Clarifications of requirement	7.2.0	7.3.0	SEC7-2GGBA
2006-03	SP-31	SP-060056	0087	-	A	GBA keys handling and UICC presence detection	7.2.0	7.3.0	TEI
2006-03	SP-31	SP-060049	0089	-	A	Clarify the confusion of the useof NAF-ID and FQDN	7.2.0	7.3.0	(SEC1) (GAAext)
2006-03	SP-31	SP-060061	0090	-	F	key derivation clarifications	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-03	SP-31	SP-060061	0091	-	F	Use of SIM for a Ua application	7.2.0	7.3.0	SEC7-GAA2 (GAAExt)
2006-06	SP-32	SP-060424	0093	1	F	UICC removal	7.3.0	7.4.0	TEI
2006-06	SP-32	SP-060380	0095	-	A	Minimum lifetime of Keys	7.3.0	7.4.0	TEI7
2006-09	SP-33	SP-060500	0096	-	F	Clarification of notes	7.4.0	7.5.0	SEC7
2006-09	SP-33	SP-060500	0097	-	F	Corrections of some incorrect reference	7.4.0	7.5.0	GAA2
2006-09	SP-33	SP-060493	0099	-	A	Sending FQDN over Zn	7.4.0	7.5.0	SEC1-SC
2006-09	SP-33	SP-060496	0101	-	A	Correction to the UICC selection procedure in GBA	7.4.0	7.5.0	TEI6
2006-12	SP-34	SP-060810	0103	1	F	Addition of note on out-of-order case for BSF	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060810	0104	1	F	Using pseudonyms over Zn to ensure service continuity	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060810	0105	1	C	UICC application selection for service continuity	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060810	0106	1	F	Two NAF applications having the same NAF keys	7.5.0	7.6.0	GAA2 (GAAExt)
2006-12	SP-34	SP-060801	0110	1	F	Correction of Requirements on GBA keys	7.5.0	7.6.0	SEC7-GAA2
2006-12	SP-34	SP-060811	0111	-	C	Suppression of NAF Group attribute within USS on Zn and Zn' interfaces	7.5.0	7.6.0	SEC7-GAA2
2006-12	SP-34	SP-060809	0112	1	F	Addition of text regarding the use of the NDS authentication framework	7.5.0	7.6.0	NDSAFTLS
2007-03	SP-35	SP-070160	0113	1	F	Key deletion method for applications	7.6.0	7.7.0	SEC7-GAA2
2007-03	SP-35	SP-070160	0114	2	F	Encode problems in HTTP digest AKA authentication between UE and BSF	7.6.0	7.7.0	GAA2
2007-03	SP-35	SP-070160	0116	1	F	GBA key names clarification	7.6.0	7.7.0	SEC7-GAA2
2007-03	SP-35	SP-070160	0118	-	D	Correction of Note in Annex H	7.6.0	7.7.0	SEC7-GAA2
2007-03	SP-35	SP-070146	0120	1	A	Clarification on NAF_Id coding	7.6.0	7.7.0	TEI6
2007-03	SP-35	SP-070147	0122	-	A	Clarification of mapping of GUSS to IMPIs and IMSIs	7.6.0	7.7.0	SEC7-GAA2
2007-06	SP-36	SP-070338	0123	1	B	Details of HLR - BSF reference point	7.7.0	7.8.0	GAAExt
2007-06	SP-36	SP-070327	0124	-	F	Clarifying the terms 2G and 3G for GBA	7.7.0	7.8.0	2G_GBA
2007-06	SP-36	SP-070327	0126	1	A	GBA NAF Keys storage policy in the UICC	7.7.0	7.8.0	SEC1-SC
2007-06	SP-36	SP-070340	0108	3	B	Introduction of temporary identifier for bootstrapping procedure	7.8.0	8.0.0	TEI8
2007-09	SP-37	SP-070594	0128	4	A	Correction to HLR - BSF reference point	8.0.0	8.1.0	SEC7-GAA2

2007-12	SP-38	SP-070792	0129	1	D	Addition of information for developers on B-TID uniquenss	8.1.0	8.2.0	TEI8
2007-12	SP-38	SP-070785	0132	1	A	2G GBA Certificate Management	8.1.0	8.2.0	TEI8
2007-12	SP-38	SP-070787	0134	1	A	Usage of OMA References – Update of Reference	8.1.0	8.2.0	TEI8
2008-03	SP-39	SP-080141	0138	1	A	Simultaneous handling of Zh' and Zh in a BSF	8.2.0	8.3.0	SEC7-GAA2
2008-03	SP-39	SP-080143	0136	-	F	Move Manual TLS certificate handling	8.2.0	8.3.0	TEI8
2008-09	SP-41	SP-080143	0139	1	F	Zh and Zh' intra-operator domain reference points	8.3.0	8.4.0	TEI8
2008-12	SP-42	SP-080744	0140	-	F	Add FC number space value allocations and clarification on length parameter	8.4.0	8.5.0	TEI8
2008-12	SP-42	SP-080744	0141	-	F	Using Unicode Standard Normalization Form when encoding using UTF-8	8.4.0	8.5.0	TEI8
2009-03	SP-43	SP-090137	0142	-	F	Add FC number space value allocations for HSPA SRVCC	8.5.0	8.6.0	TEI8
2009-06	SP-44	SP-090420	0142	1	F	Ua security protocol identifiers for IMS based MBMS	8.6.0	8.7.0	TEI8
2009-06	SP-44	SP-090420	0143	1	F	Clarify sending of MSISDN in Zn	8.6.0	8.7.0	TEI8
2009-06	SP-44	SP-090276	0142	-	C	Modification of References	8.7.0	9.0.0	TEI9
2009-06	SP-44	SP-090276	0144	1	B	GBA DIAMETER based Zn reference point to support TLS	8.7.0	9.0.0	TEI9
2009-09	SP-45	SP-090524	0143	-	F	FC value allocation for GPL	9.0.0	9.1.0	eGBAPush
2009-09	SP-45	SP-090524	0145	-	C	Introducing Ua security protocol Id for GPL	9.0.0	9.1.0	eGBAPush
2009-12	SP-46	SP-090820	0146	-	A	KDF clarification	9.1.0	9.2.0	eGBAPush
2009-12	SP-46	SP-090822	0148	-	F	Ua security protocol identifier for IMS media plane security	9.1.0	9.2.0	MEDIASEC
2010-06	SP-48	SP-100361	0149	1	F	Deprecation of SHA-1	9.2.0	9.3.0	TEI9
2010-10	SP-49	SP-100482	0150	1	C	Unification of TLS and certificate references in TS 33.220 with TS 33.310	9.3.0	10.0.0	TEI10
2011-09	SP-53	SP-110563	0151	-	F	Reintroduction of lost reference	10.0.0	11.0.0	Sec11
2011-12	SP-54	SP-110848	0153	1	F	Correction of Allowed TLS Ciphersuite Identifiers in Annex H.3	11.0.0	11.1.0	Sec11
2012-03	SP-55	SP-120033	0156	3	B	SIP Digest-based GBA scope and terminology updates	11.1.0	11.2.0	GBA-ext
			0159	2	B	GBA extension for re-use of SIP Digest credentials			
		SP-120039	0157	-	F	Update of TLS extensions version			Sec11
			0158	-	F	Correction of BSF and bootstrapping requirements in 2G GBA			
			0160	1	D	GBA terminology issues			
		SP-120037	0164	-	A	Correction of misimplementation of Change Requests on KDF FC value allocations			TEI8
2012-06	SP-56	SP-120340	0165	2	F	Introduction of auth-int in GBA_Digest	11.2.0	11.3.0	GBA-ext
2012-06	SP-56	SP-120341	0166	-	F	Correction of TLS Extensions References to point toTS 33.310	11.2.0	11.3.0	SEC11
2012-06	SP-56	SP-120341	0167	-	F	Correction of phrase describing Zn procedure	11.2.0	11.3.0	SEC11
2012-06	SP-56	SP-120340	0168	-	F	NAF specific key derivation in GBA_Digest	11.2.0	11.3.0	GBA-ext
2012-06	SP-56	SP-120340	0169	-	F	TMPI (temporary identity) support in GBA_Digest	11.2.0	11.3.0	GBA-ext
2012-09	SP-57	SP-120605	0170	1	F	Correction of description of “Bootstrapping Initiation”	11.3.0	11.4.0	SEC11
2013-06	SP-60	SP-130255	0174		F	Removal of editor's note Release 11 - 33.220	11.4.0	11.5.0	FS_SSO_APS