OMA GBA Profile

Candidate Version 1.1 – 30 Nov 2010

Open Mobile Alliance

OMA-TS-GBA_Profile-V1_1-20101130-C

Contents

Figures

1. Scope

This specification defines an OMA profile of the Generic Bootstrapping Architecture (GBA) specified in 3GPP/3GPP2 and also an OMA profile of GBA Push specified in 3GPP. GBA specifies an architecture where operator controlled smart cards, i.e., SIMs for GSM, and USIMs/ISIMs for UMTS and UICCs/R-UIMs/CSIMs for 3GPP2) can be used to bootstrap a short term security association between a client and a server in the network. This short term security association can be used between any client and any server in the network to secure the connection between them. GBA Push also specifies a similar architecture, where the NAF initiates the bootstrapping process, but the scheme applies only to USIM/ISIM.

GBA is often used in OMA Enabler Specifications, and existence of several options in the GBA specifications may lead to interoperability problems in GBA usage in OMA Enablers. In addition, requiring implementations of some options in the TLS protocol for OMA enablers can increase the level of security compared to only implementing the mandatory features in the TLS specifications. OMA Workgroups developing enabler specifications are recommended to use the OMA Profile of GBA.

2. References

2.1 Normative References

[OMNA] "Open Mobile Naming Authority", URL: http://www.openmobilealliance.org/Tech/OMNA.aspx

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2617] "HTTP Authentication: Basic and Digest Access Authentication", IETF RFC 2617, June 1999, URL: http://www.ietf.org/rfc/rfc2617.txt

[RFC 3324] "Short Term Requirements for Network Asserted Identity", November 2002, http://www.ietf.org/rfc/rfc3324.txt

[RFC4279] "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, P. Eronen, et al, December 2005

URL: http://www.ietf.org/rfc/rfc4279.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[S.S0109] "Generic Bootstrapping Architecture (GBA) Framework", 3GPP2 S.S0109-0, Version 1.0, 30th March 2006, URL: http://ww.3gpp2.org

[S.S0114] "Security Mechanisms using GBA", 3GPP2 S.S0114-0, Version 1.0, 30th March 2006, URL: http://ww.3gpp2.org

[TS24109] "Bootstrapping interface (Ub) and network application function interface (Ua); Protocol details", 3GPP TS 24.109, URL: http://www.3gpp.org/

[TS29109] "Zh and Zn Interfaces based on the Diameter protocol; Stage 3", 3GPP TS 29.109, URL: http://www.3gpp.org/

[TS33220] "Generic Bootstrapping Architecture (GBA)", 3GPP TS 33.220, URL: http://www.3gpp.org/

[TS33220-7] "Generic Bootstrapping Architecture (GBA)", 3GPP TS 33.220, URL: http://www.3gpp.org/

[TS33221] "Support for Subscriber Certificates", 3GPP TS 33.221, URL: http://www.3gpp.org/

[TS33222] "Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)", 3GPP TS 33.222, URL: http://www.3gpp.org/

[TR33223] "Generic Bootstrapping Architecture (GBA) Push Function",3GPP TS 33.223, URL: http://www.3gpp.org/

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org/

[SEC_CF AD] "Common Security Functions Architecture", OMA-AD-SEC_CF-V1_1

URL: http://www.openmobilealliance.org

[SEC_CF TLS] "Common Security Functions Architecture, OMA TLS profile", OMA-TS-TLS_Profile-V1_1

URL: http://www.openmobilealliance.org

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Definitions

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	Using 3GPP or 3GPP2 authentication infrastructure the Client and the BSF generate joint session keys. This key material can be used by the NAF (but this is not part of the actual bootstrapping).
Bootstrapping Server Function	BSF is hosted in a network element under the control of an MNO. BSF, HSS, and Clients 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 Clients, for example, for authentication purposes.
Client	A 3GPP UE (User Equipment) or 3GPP2 MN (Mobile Node)
Network Application Function	NAF is hosted in a network element. GBA may be used between NAFs and Clients for authentication purposes, and for securing the communication path between the Client and the NAF.
Push-message	This is a message that is sent on a Ua-reference point from the NAF to the UE and has applied GBA keys that were bootstrapped via the Upa-reference point.
Push-NAF	A NAF authorized for using GBA-Push

3.3 Abbreviations

3GPP	Third Generation Partnership Project
3GPP2	Third Generation Partnership Project 2
AKA	Authentication and Key Agreement
BSF	Bootstrapping Server Function
C-SIM	CDMA Subscriber Identity Module
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
GBA_ME	ME based GBA
GBA_U	GBA with UICC-based enhancements
GPI	GBA Push Info
HLR	Home Location Registry
HSS	Home Subscriber System
HTTP	Hyper Text Transport Protocol
HTTPS	HTTP over TLS

IMS	IP Multimedia Service
ISIM	IP Multimedia Services Identity Module
MNO	Mobile Network Operator
NAF	Network Application Function
OMA	Open Mobile Alliance
PSK-TLS	Pre-Shared Key TLS
R-UIM	Removable UIM
SCR	Static Conformance Requirements
SIM	Subscriber Identity Module
TLS	Transport Layer Security
UICC	Universal Integrated Circuit Card
UIM	User Identity Module
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
USS	User Security Settings

4. Introduction

GBA (Generic Bootstrapping Architecture) (see [TS33220], [S.S0109]] provides a secure and reliable method to bootstrap a security association between a client and a server. GBA uses long term security associations that are stored in a tamper resistant module in the client, and in the central network element (e.g., a Home Location Registry (HLR) or in the Home Subscriber System (HSS) of the mobile network operator (MNO). The tamper resistant device is typically a smart card or a UICC (Universal Integrated Circuit Card). Based on this long term security association, a short term server specific security associations, i.e., GBA credentials, a Ks, are created during a bootstrapping procedure between the client and a dedicated network element Bootstrapping Server Function (BSF) controlled by the MNO. The GBA credentials can be used between the client and the server functioning as a Network Application Function (NAF) provided that the operator of this server has an agreement with a MNO and has access to the BSF.

GBA (Generic Bootstrapping Architecture) Push function [TS33223] allows a NAF to initiate establishment of a shared Security Association (SA), a NAF SA, between itself and a UE. This is done by the NAF pushing all information, the so called GBA-Push-Info (GPI), needed for the UE to set-up the SA. The key in this SA is a NAF-key and the GPI is requested from the BSF. GBA-Push utilizes a so called Disposable-Ks model where the Ks is only used once to derive a single set of NAF-keys.

The client and the server may use the short term shared SA NAF_key as they want, e.g., using it as username and password in HTTP Digest authentication, or as pre-shared secret in PSK-TLS.

This specification aims to provide a common profile of GBA and GBA Push that can be used by all the OMA Enablers. The intention is to create a secure and interoperable GBA that can be re-used without the need to define the profile for GBA separately in each OMA Enabler specifications.

Developers of OMA Enablers who wish to use GBA are recommended to use this specification to define the requirements for their GBA usage.

4.1 Version 1.0

The TS-GBA_Profile v1.0 is to support shared secret key negotiation between the client and the server operating over the protocol HTTP.

4.2 Version 1.1

The TS-GBA_Profile v1.1 is to improve SEC_CFv1.1 to provide security protection for Push services and to support services operating over the protocol SIP.

5. OMA GBA Profile

OMA GBA Profile is based on 3GPP GBA [TS33220] , 3GPP GBA Push [TS33223], 3GPP2 GBA [S.S0109] and relevant specifications. All OMA GBA Profile compliant implementations MUST also conform to the relevant specifications of at least one of the following groups:

- The 3GPP GBA specification family [TS33220], [TS24109], and [TS29109].
- The 3GPP GBA Push specification [TS33223].
- The 3GPP2 GBA specification family [S.S0109] and [S.S0114].

This specification profiles a particular implementation of 3GPP GBA/GBA Push, 3GPP2 GBA and other relevant specification that can be used with GBA such Digest authentication [RFC2617] and PSK-TLS [RFC4279].

Normative text included this section MUST be considered as additions to the existing baseline GBA and related specifications. All terminology in this specification MUST be taken in the context of GBA and related specifications.

5.1 Supported Bootstrapping Mechanisms

5.1.1 Bootstrapping Mechanisms supported by a Server

The Server that wishes to support clients on a 3GPP handset MUST be able to use GBA credentials established using any of the following bootstrapping mechanisms:

- GBA and GBA Push based on GBA_ME as specified in 3GPP [TS33220] and [TS33223] respectively.
- GBA and GBA Push based on GBA_U as specified in 3GPP [TS33220] and [TS33223] respectively.
- GBA based on 2G GBA as specified in 3GPP [TS33220-7 and onwards].

The Server that wishes to support clients on a 3GPP2 handset MUST be able to use GBA credentials established using any of the following bootstrapping mechanisms:

- GBA based on GBA_ME as specified in 3GPP2 [S.S0109]
- GBA based on GBA_U as specified in 3GPP2 [S.S0109]
- GBA based on 2G GBA as specified in 3GPP2 [S.S0109].

Note: From the Server's perspective, for GBA, the interactions with the Client for 3GPP2 GBA_ME and 3GPP2 GBA_U [S.S0109] are identical to the interactions for 3GPP GBA_ME and 3GPP GBA_U [TS33220] respectively.

5.1.2 Bootstrapping Mechanisms supported by a Client

The Client on a 3G Handset for 3GPP network:

- MUST support GBA based on GBA_ME and GBA_U as specified in 3GPP [TS33220],
- SHOULD support GBA Push based on GBA_ME and GBA_U as specified in 3GPP [TS33223],
- MAY optionally support GBA based on 2G GBA as specified in 3GPP [TS33220-7 on onwards].

The Client on a 3G Handset for 3GPP2 network:

- MUST support GBA based on GBA_ME and GBA_U using AKA as specified in 3GPP2 [S.S0109]; or MUST support GBA based on GBA_ME and GBA_U using MN-AAA Key as specified in 3GPP2 [S.S0109],
- MAY optionally support GBA based on 2G GBA using CAVE as specified in 3GPP2 [S.S0109].

5.2 Specific features related to GBA implementation

5.2.1 Protocol identifier

A Ua security protocol identifier has been introduced to enable key separation in GBA. It consists in a string of 5 octets:

- the 1st octet concerns the organization, OMA has been assigned the value 0x03, which has to be used by any OMA enabler.
- the 4 following octets are specific to the security protocol, under the responsibility of the organization.

For instance for an OMA enabler implementing a shared key TLS mechanism, the protocol identifier follows the format:

Organization identifier	Security protocol identifier
0x03	Shared key TLS ID

The security protocol identifier has to be updated according to the OMA enabler needs. Such protocol identifier can be used to derive different keys with specific purpose instead of specifying complex mechanism (ex:integrity protection of messages sent by the NAF to the device).

All such uses of the OMA protocol identifiers will be recorded in the OMNA registry [OMNA] established for these identifiers. In addition, the OMNA registry will capture, where disclosed, OMA Enabler use of identifiers based on other organization codes.

The main goal is to be able to generate two different Ks_NAF/ Ks_(ext/int)_NAF keys from the same Ks, for different purposes corresponding to the same B-TID. The protocol ID feature could be used since it is included in the NAF ID value. The following specification can be envisaged:

NAF ID = domain name || protocol ID.

Where:

- protocol ID 1: derivation of Ks will lead to the generation of Ks_NAF1/ Ks_(ext/int)_NAF1 for integrity purpose.
- protocol ID 2: derivation of Ks will lead to the generation of Ks_NAF2/ Ks_(ext/int)_NAF2 for TLS purpose.

5.2.2 GBA Keying Models

The keying model in GBA Push differs from that of GBA.

For GBA, in case the shared secret key Ks_NAF/ Ks_(ext/int)_NAF is not valid anymore (ex: if the key lifetime has expired), and depending on OMA enabler properties the UE SHALL initiate a new GBA procedure to provision a valid value for Ks_NAF/ Ks_(ext/int)_NAF.

Regarding GBA_U implementation the detection of a new USIM/ISIM within the device MUST lead to a new GBA procedure to generate a new shared secret for the new USIM/ISIM. This MUST lead to the suppression of all keys Ks and Ks_ext_NAF within the device. This assumes that the mechanism to detect a new USIM/ISIM is qualified as reliable enough.

For GBA Push, the Disposable-Ks model states that a Ks is only used once to derive a single set of NAF-keys (and other keying material used to protect the GPI during transport. After the NAF-key derivation, the Ks is erased or its further use is denied implicitly, which means that there will be no generally usable Ks established.

5.2.3 NAF Domain name

The NAF MUST have Fully Qualified Domain Name.

5.2.4 GBA Push Info (GPI) Transport

The transport method of GPI from a NAF to a UE is not standardized. Possible transport methods are SMS, MMS, SIP MESSAGE, UDP or broadcast. The NAF needs to know the message transport addresses to use for the chosen transport method.

When the underlying transport doesn't provide reliability, resending of messages is a standard method to get "reliability" for delivery over unreliable channels like e.g. SMS or broadcast.

5.3 Interoperator GAA

This section specifies how interoperator GAA works both between different operators utilizing 3GPP GBA (and 3GPP2 GBA). In GBA, the Client always bootstraps with the home BSF. In GBA and GBA Push, interoperator GAA ensures that if NAF operates in a different network than subscriber's home network, the "foreign" NAF is able to request GBA credentials from subscriber's home BSF across operator boundaries. Example of how Interoperator GAA is setup is described in informative annex *K* of 3GPP TS 33.220 Release-7 [TS33220-7 and onwards].

Access control based on the use of USSs can be implemented (cf. Appendix C).

The use case corresponds to a NAF located within a visited network. Thus it has to be ensured that application specific profiles can be considered within a multiple operator's environment so that the visited network is able to control whether the subscriber is allowed to access the service in the visited network.

5.4 GBA Usage in Digest Authentication

If the Client and the Server use GBA in Digest authentication [RFC2617], the Client and the Server MUST comply the procedures defined in [TS33220] and [TS24109].

5.5 GBA Usage in HTTPS with Digest Authentication

TLS profile MUST be based on OMA TLS Profile, which is part of SEC_CF.

If the Client and the Server use GBA in HTTPS with Digest authentication, the Client and the Server MUST comply with the procedures defined in [TS33222] and [TS24109]. The Server MUST comply with the procedures defined in [TS29109].

5.6 GBA Usage in PSK TLS

TLS profile MUST be based on OMA TLS Profile, which is part of SEC_CF.

If the Client and the Server use GBA in PSK TLS, the Client and the Server MUST comply with the procedures defined in [TS33222] and [TS24109] or [S.S0114].

A Client that wishes to indicate support for GBA in PSK TLS MUST comply with the procedures defined in

- [TS33222] and [TS24109] if the Client is a 3GPP handset.
- [S.S0114] if the Client is a 3GPP2 handset.

A Server that wishes to indicate support for GBA in PSK TLS MUST comply with the procedures defined in [TS33222], [TS24109] and [S.S0114]. Note that the 3GPP2 specification for GBA in TLS PSK [S.S0114] was written so as to agree with [TS33222] and [TS24109].

5.7 GBA over SIP

OMA Enablers that rely on SIP/IP Core for user plane data and that want to support clients on a 3GPP handset MUST be able to use GBA credentials using the procedures described below. It is assumed that only SIP is used for communication between the client and NAF.

5.7.1 SIP-based Authentication

There are two procedures to authenticate a client to a NAF-enabled enabler: complete and optimized.

For both procedures, it is assumed that:

- The client has successfully executed a GBA bootstrapping with the BSF as defined in [TS33220].
- A Ua security protocol identifier has been registered for usage by the OMA enabler implementing this NAF functionality

5.7.1.1 Complete GBA/SIP Procedure

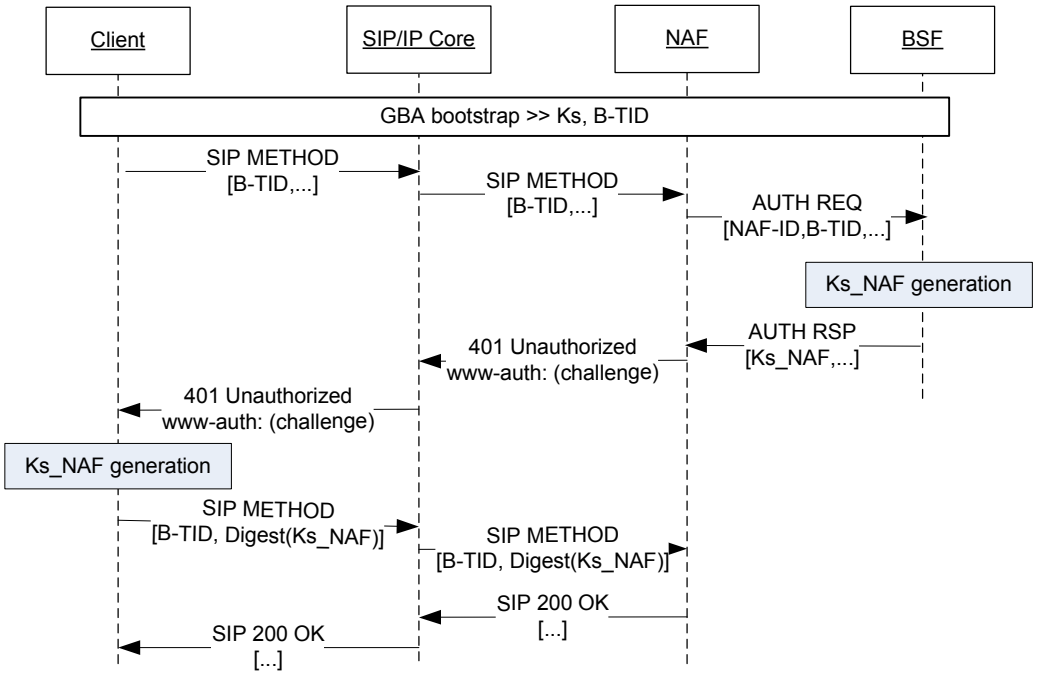

Figure 1. Complete GBA/SIP-based authentication

The complete procedure is illustrated in figure 1 and works in more detail as follows:

- The client uses an appropriate SIP method (e.g. INVITE or SUBCRIBE) to the NAF via the SIP/IP core. The SIP method MUST include the B-TID in the SIP body. The client MAY include other application specific information using other schemas.
- In addition to the B-TID, the NAF receives the identity of the client in the headers of the SIP method.
- Using GBA procedures, the NAF constructs the NAF-ID using the allocated Ua security protocol identifier to obtain the NAF key from the BSF.

- The NAF authenticates the client. The figure illustrates this process for the case of SIP Digest authentication, but other authentication methods are possible based on the shared secret NAF key. The Ua security protocol identifier that shall be used for SIP Digest is defined in OMNA Registry [OMNA] under *GBA Protocol Identifiers.*

5.7.1.2 Optimized Procedure

This procedure assumes that:

- The NAF can receive in SIP headers the IP address and the asserted identity corresponding to the client and thus explicit authentication is not needed
- The SIP IP/Core is a is considered a set of Trust Domains of SIP servers that are trusted to handle Network Asserted Identity according to RFC 3324. A trust domain has been protected using appropriate security procedures as for example NDS/IP defined in [TS33210].

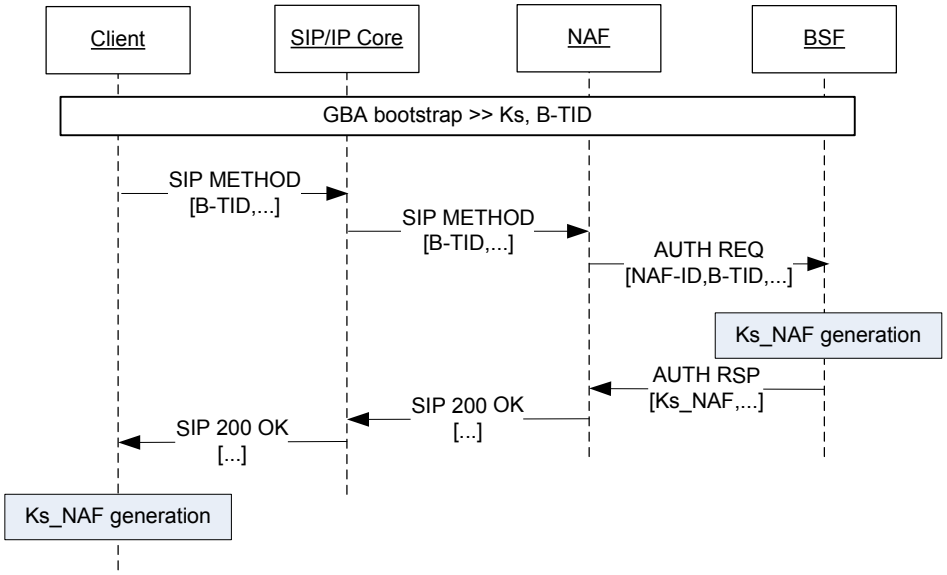

Figure 2. Optimized GBA/SIP-based authentication

The optimized procedure is illustrated in figure 2 and exploits the underlying SIP IP/core Trust Domain to shortcut explicit authentication.

Note that due to optimization, the Ks_NAF is not used for authentication, but it can still be used by the NAF application for other security purposes, e.g. further key derivation, integrity or confidentiality.

5.7.2 XML Schema for B-TID

The following example shows an XML document that carries the bootstrapping transaction within the body of a SIP METHOD from the client to the NAF.

```
<?xml version="1.0" encoding="UTF-8"?>
<gba:bootstrappingTransactionIdentifier xmlns:gba="urn:oma:xml:seccf:gba:1">
     <btid>Yu9o/TY2IQ0k6zKjcHGSFw==@bsf.example.com</btid>
</gba:bootstrappingTransactionIdentifier>
```

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
Approved Versions OMA-TS-GBA_Profile-V1_0	02 Sep. 2008	Status changed to Approved by TP OMA-TP-2008-0321-INP_SEC_CF_V1_0_ERP_for_Final_Approval

A.2 Draft/Candidate Version 1.1 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-GBA_Profile-V1_1	25 Dec 2009	All	Incorporates input to committee: OMA-ARC-SEC-2009-0086R01-INP_Baseline_of_SEC_CF1.1_TS_Development
	15 Apr 2010	All	OMA-ARC-SEC-2010-0027-CR_GBA_over_SIP, OMA-ARC-SEC-2010-0031R01-CR_GBA_Push_Profile
	01 Sep 2010	All	OMA-ARC-SEC-2010-0071R01-CR_GBA_over_SIP.doc; OMA-ARC-SEC-2010-0078R01-CR_GBA_Profile_Introduction.doc.
	03 Oct 2010	2, 3	OMA-ARC-SEC-2010-0094-CR_CR_GBA_Profile_Bibliography_Abbreviations
	04 Oct 2010	All	Editorial fixes: 2010 copyright Removed empty rows in tables Fixed History table Added Figures in Contents
	29 Oct 2010	2.1, 5.2.4	Incorporates the input OMA-ARC-SEC-2010-0107-CR_CONR_SEC_CF1.1_TS_GBA_Profile_D001_D002
Candidate Version OMA-TS-GBA_Profile-V1_1	30 Nov 2010	All	Status changed by TP: OMA-TP-2010-0472-INP_SEC_CF_V1_1_ERP_for_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for GBA Client

Item	Function	Reference	Requirement
GBA-C-001-M	OMA GBA implementation conform to [TS33220] & [TS33223]	Section 5.	
GBA-C-002-M	Protocol identifier	Section 5.2.1	

B.2 SCR for NAF Server

Item	Function	Reference	Requirement
GBA-S-001-M	OMA GBA implementation conform to [TS33220] & [TS33223]	Section 5.	
GBA- S-002-M	Protocol identifier	Section 5.2.1	

Appendix C. <Additional Information>

Editor's Note: the use of GUSS should be based on [TS 33.220-7] specifications..

Editor's Note: the appropriate specifications for 3GPP2 GBA User Security Settings are not yet specified. When specified, the appropriate text should be included.

The GUSS enables the implementation of a local policy enforcement within the BSF.

GBA User Security Settings (Informative)

C.1 GUSS - main features

Here are the main points of the GBA User Security Settings.

There is one GUSS per subscription (USIM or ISIM), identified by the private identity (IMPI format).

A GUSS contains:

- Subscriber information for the BSF : e.g. UICC type = GBA_U
- A set of USS for the applications that the subscriber may use.
- USS may be transferred by the BSF to the NAF with the key material

- USS contains application specific information
  - USS is identified by GSID (GAA Service Id)
  - Authentication part : list of user identities
  - Authorization part : list of user permission flags whose signification depends on the application type

Upon reception of the Bootstrap transaction ID from the user the NAF retrieves authentication information and the USS from the BSF.

Upon reception of the request from the NAF, the BSF checks whether a local policy is associated for this NAF, and whether the NAF is authorized to receive the USS corresponding to the GSID in the incoming request. If the control is valid, then the BSF derives the NAF specific shared keys, and sends them to the NAF with the available USSs. From specific application criteria (specifically the authorization part) the NAF decides to valid the access control or not.

It is important to notice that the NAF can indicate to the BSF for which application it requires a USS. This enables granularity for the definition of authorizations. USSs may contain NAF specific information such as authorization flags etc.which enable to take the decision about allowing the access to the service or not.

The GUSS format follows an XMLschema definition. Following is an extract from [TS29.109] annex A:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="guss-schema-of-3gpp-gaa"
    xmlns:tns="guss-schema-of-3gpp-gaa"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

  <!-- This import brings in the XML language attribute xml:lang-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
      schemaLocation="http://www.w3.org/2001/xml.xsd"/>

  <xs:complexType name="tExtension">
    <xs:sequence>
      <xs:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
```

```
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tGUSSExtension">
    <xs:sequence>
      <xs:element name="timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="Extension" type="tExtension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tUSSExtension">
    <xs:sequence>
      <xs:element name="keyChoice" type="xs:string" minOccurs="0" />
      <xs:element name="Extension" type="tExtension" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>

  <!-- The whole user's GBA specific data set  -->
  <xs:complexType name="guss">
    <xs:sequence>
       <xs:element ref="bsfInfo" minOccurs="0"/>
       <xs:element ref="ussList"/>
       <xs:element name="Extension" type="tGUSSExtension" minOccurs="0"/>
       <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
  </xs:complexType>

  <!-- BSF specific information element -->
  <xs:complexType name="bsfInfo">
    <xs:sequence>
      <xs:element name="uiccType"  type="xs:string"  minOccurs="0" />
      <xs:element name="lifeTime"  type="xs:integer" minOccurs="0" />
      <xs:element name="Extension" type="tExtension" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!--List of all users individual User Security Settings -->
  <xs:complexType name="ussList">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
       <xs:element ref="uss"/>
       <xs:element name="Extension" type="tExtension" minOccurs="0"/>
       <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!-- User Security Setting data -->
  <xs:complexType name="uss">
    <xs:sequence>
      <xs:element ref="uids"/>
      <xs:element ref="flags"/>
      <xs:element name="Extension" type="tUSSExtension" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
```

```
    </xs:sequence>
    <xs:attribute name="id"   use="required" type="xs:string"/>
    <xs:attribute name="type" use="required" type="xs:int"/>
    <xs:attribute name="nafGroup" use="optional" type="xs:string"/>
  </xs:complexType>

  <!-- User Public Identities for authentication -->
  <xs:complexType name="uids">
    <xs:sequence minOccurs="1" maxOccurs="unbounded">
      <xs:element name="uid"  type="xs:string"/>
      <xs:element name="Extension" type="tExtension" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <!-- GAA Application type specific Authorization flag codes -->
  <xs:complexType name="flags">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="flag"  type="xs:int"/>
      <xs:element name="Extension" type="tExtension" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

- Bsfinfo field indicates a specific key lifetime: if an application requires a specific update policy regarding keys, then this parameter should be used accordingly. However the value is common for all keys generated for the user, so a unique value is defined for all services used the user.
- Attribute "type" in the element USS is linked with a GAA service type code to indicate interpretation/meaning/coding of the GAA service type specific data. An application may need to define its own set of authorization flags, and they can be specified in the future. Then the access to the service is allowed only if the client's USS contains that flag. So a service code is directly linked to an application profile.
- Attribute "flag" is linked to a user's authorization flag. It is referred within the HSS and is related to a GAA service type (cf. attribute type in the parent USS element). The values of these flags determine if the NAF have permission to give the corresponding service, otherwise not.

  For example [TS29.109] defines PKI portal (used to request a certificate), with the following set of flags:

  - Authentication allowed
  - Non-repudiation allowed

  Then the USS of the subscriber for this particular application shall contain the flags listed above to be able to use the service PKI portal. This enables to generate certificates with these special permissions:

  - certificate to handle authentication
  - certificate to handle non-repudiation
  - certificate to handle non-repudiation and authentication

  OMA enablers can then define their own flags, conditioning the access to the service.

- Attribute "KeyChoice" indicates which GBA credentials should be used either "ME-based-key" i.e., Ks_ NAF or Ks_ext_NAF shall be used, or "UICC-based-key", i.e., Ks_int_NAF shall be used or " ME-UICC-based-keys", i.e., Ks_ext_NAF or Ks_int_NAF can be used. This parameter should be considered for any application.

Note: The GUSS is stored within the HSS and is retrieved by the BSF at user authentication (for example the NAF requests the NAF specific key(s) from the BSF). If the GUSS is updated in the HSS, then the updates will be taken into account once the BSF retrieves the authentication vector from the HSS during the bootstrap procedure[1]. Then this change is transmitted onto the NAF once it retrieves the USS from the BSF. A study case by case has to determine whether this update has to be taken into account immediately, or during the next subscriber's authentication procedure.

C.2 GUSS - main use cases

A GUSS contains information that may lead the NAF to validate the access to the service or not. The following lists the allowed types of parameters in the application specific USS:

- key selection (indicates the use of GBA_U).
- identification information (phone number, or email address etc..)
- authorization information

Different GUSS can be handled by different applications.

Such feature should be used as most as possible by OMA enablers, avoiding the implementation of useless infrastructure, additional software code.

Note

- Support of GUSS elements should be supported without need for standardization.
- To avoid any confusion the HSS should remove any USS related to unused services, or once the end of a service subscription is reached.

C.3 Requirements on the NAF

Based on the USS key selection value, the NAF can be configured in order to restrict the access to the service depending on which key has to be used: Ks_ext_NAF or Ks_int_NAF respectively for GBA_ME and GBA_U implementations. The USS indication with the keyChoice value is the reference for this information.

This feature should be used as most as possible, if the OMA enabler has specific restrictions regarding GBA implementation. For instance an OMA enabler can restrict the access to HTTPS capable clients only on the UICC, with a GBA_U based implementation with the value Ks_int_NAF.

C.4 Roaming use case

The GUSS based access control is local to the home operator network.

In a roaming use case, a visited NAF requires the BSF of the home network to retrieve user information. Furthermore for this specific use case, the following local policy could be defined: to prevent the misuse of services that require authentication, the BSF should require one (or more) application specific USS to be contained in the user's GUSS for a visited NAF, and the request should be rejected in the cases where these conditions are not fulfilled.

[1] This feature is enabled only if the BSF implements the timestamp option.