3GPP TSG CN Meeting #27 **NP-050082**
9th - 11th March 2005. Tokyo, Japan.

Source: **TSG CN WG1**

Title: **CR to Rel-6 WI "SEC1-SC" for TS 24.109**

Agenda item: **9.3**

Document for: **APPROVAL**

This document contains **CR on Rel-6 Work Item "SEC1-SC"**, that has been agreed by TSG CN WG1 CN#37 meeting and forwarded to TSG CN Plenary meeting #27 for approval.

TDoc #	Tdoc Title	Spec	CR #	Rev	CAT	C_Version	WI	Rel
N1-050357	Editorial corrections	24.109	11	1	F	6.1.0	SEC1-SC	Rel-6

3GPP TSG-CN1 Meeting #37
Sydney, Australia, 14-18 February 2004

Tdoc N1-050357

CR-Form-v7.1

CHANGE REQUEST

⌘ **24.109** **CR** **11** ⌘ **rev** **1** ⌘ Current version: **6.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Editorial corrections
Source:	⌘	Nokia
Work item code:	⌘	SEC1-SC
Date:	⌘	16/2/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Editorial fixes
Summary of change:	⌘	Editorial fixes in the document.
Consequences if not approved:	⌘	Unclear parts in specification

Clauses affected:	⌘	2, 3, 4.2, 5.2.1, 5.3.1, 5.3.2, 5.3.3.1, A.3, B.3, C.1

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

⌘

Other comments: ⌘

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 33.220: "Generic Authentication Architecture (GAA); Generic bootstrapping architecture".

[2] 3GPP TR 33.919: "Generic Authentication Architecture (GAA); System description".

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

[4] 3GPP TS 33.221: "Generic Authentication Architecture (GAA); Support for Subscriber Certificates".

[5] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

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

[7] 3GPP TS 23.003: "Numbering, addressing and identification".

[8] IETF RFC 3023: "XML Media Types".

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

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

[11] IETF RFC 2246: "The TLS Protocol Version 1.0".

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

[13] 3GPP TS 24.228: "Signalling flows for the IP multimedia call control based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

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

[15] IETF draft-ietf-tls-psk-01: "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)".

[16] PKCS#10 v1.7: "Certification Request Syntax Standard".

NOTE: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-10/pkcs-10v1_7.pdf

[17] WAP Forum: "WPKI: Wireless Application Protocol; Public Key Infrastructure Definition"

NOTE: http://www1.wapforum.org/tech/documents/WAP-217-WPKI-20010424-a.pdf.

[18] IETF RFC 3546: "Transport Layer Security (TLS) Extensions".

[19] Open Mobile Alliance: "ECMAScript Crypto Object"

NOTE: http://www.openmobilealliance.org.

[20] Open Mobile Alliance: "WPKI"

NOTE: http://member.openmobilealliance.org/ftp/public_documents/SEC/Permanent_documents/.

[21] 3GPP TS 33.203: "3G security; Access security for IP-based services".

[22] IETF RFC 2234: "Augmented BNF for Syntax Specifications: ABNF".

3 Definitions and abbreviations

3.1 Definitions

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

Bootstrapping information: set of parameters that have been established during bootstrapping procedure
The information consists of a bootstrapping transaction identifier (B-TID), key material (Ks), and a group of application specific security parameters related to the subscriber.

Bootstrapped security association: association between a UE and a BSF that is established by running bootstrapping procedure between them
The association is identified by a bootstrapping transaction identifier (B-TID) and consists of bootstrapping information.

CA certificate~~:~~ : The Certificate Authority public key is itself contained within a certificate, called a CA certificate. The CA sign~~s~~ all certificates that it issues with the ~~its~~ private key that corresponds to the public key in the CA certificate.
~~The corresponding Certificate Authority public key is itself contained within a certificate, called a CA Certificate.~~

Delivery of CA certificate: procedure during which UE requests a root certificate from PKI portal, who delivers the certificate to the UE
The procedure is secured by using GBA.

PKI portal: certification authority (or registration authority) operated by a cellular operator

Reverse proxy: a reverse proxy is a gateway for servers, and enables one server (i.e., reverse proxy) to provide content from another server transparently, e.g., when UE's request for a particular information is received at a reverse proxy, the reverse proxy is configured to request the information from another server. The reverse proxy functionality is transparent to the UE, i.e., the UE does not know that the request is being forwarded to another server by the reverse proxy.

Root certificate: a certificate that an entity explicitly trusts, typically a self-signed CA certificate

Subscriber certificate: certificate issued to a subscriber
It contains the subscriber's own public key and possibly other information such as the subscriber's identity in some form.

Subscriber certificate enrolment: procedure during which UE sends certification request to PKI portal and who issues a certificate to UE
The procedure is secured by using GBA.

WAP Identity Module (WIM): used in performing WTLS, TLS, and application level security functions, and especially, to store and process information needed for user identification and authentication
The WPKI may use the WIM for secure storage of certificates and keys (see 3GPP TS 33.221 [4], OMA ECMAScript [19], and OMA WPKI [20] specifications).

3.2 Abbreviations

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

AKA	Authentication and Key Agreement
AP	Authentication Proxy
AS	Application Server
AUTN	Authentication Token

AUTS	Re-synchronisation Token
AV	Authentication Vector
BSF	BootStrapping Function
B-TID	Bootstrapping - Transaction IDentifier
CA	Certification Authority
CK	Confidentiality Key
DER	Distinquished Encoding Rules
FQDN	Fully Qualified Domain Name
GAA	Generic Authentication Architecture
GBA	Generic Bootstrapping Architecture
HSS	Home Subscriber System
IK	Integrity Key
IMPI	IP Multimedia Private Identity
IMPU	IP Multimedia PUblic identity
Ks	Key material
Ks_NAF	NAF specific key material
MAC	Message Authentication Code
NAF	Network Application Function
PKCS	Public-Key Cryptography Standards
PKI	Public Key Infrastructure
PSK	Pre-Shared Secret
RAND	RANDom challenge
RES	authentication Response
SQN	SeQuence Number
TLS	Transport Layer Security
UE	User Equipment
URI	Uniform Resource Identifier
URN	Uniform Resource Name
USIM	User Service Identity Module
WIM	Wireless~~WAP~~ Identity Module
WPKI	Wireless PKI
WTLS	Wireless Transport Layer Security
XRES	Expected authentication response

----------NEXT CHANGE----------

4.2 Bootstrapping procedure

The UE shall initiate the bootstrapping procedure when:

a) the UE wants to interact with a NAF and bootstrapping is required;

b) a NAF has requested bootstrapping required indication as described in subcluse 5.2.4 or bootstrapping renegotiation indication as described in subclause 5.2.5; or

~~the lifetime of the key has expired in the UE if and one or more applications are using that key.~~

c) the lifetime of the key has expired in the UE if one or more applications are using that key.

A UE and the BSF shall establish bootstrapped security association between them by running bootstrapping procedure. Bootstrapping security association consists of a bootstrapping transaction identifier (B-TID) and key material Ks. Bootstrapping session on the BSF also includes security related information about subscriber (e.g. user's private identity). Bootstrapping session is valid for a certain time period, and shall be deleted in the BSF when the session becomes invalid.

Bootstrapping procedure shall be based on HTTP Digest AKA as described in 3GPP TS 33.220 [1] and in RFC 3310 [6] with the modifications described below.

The BSF address is derived from the IMPI or IMSI according to 3GPP TS 32.220 [1].

In the bootstrapping procedure, Authorization, WWW-Authenticate, and Authentication-Info HTTP headers shall be used as described in RFC 3310 [6] with following exceptions:

a) the "realm" parameter shall contain the network name where the username is authenticated;

b) the quality of protection ("qop") parameter shall be "auth-int"; and

c) the "username" parameter shall contain user's private identity (IMPI).

NOTE: If the UE does not have an ISIM application with an IMPI, the IMPI will be constructed from IMSI, according to 3GPP TS 23.003 [7].

In addition to RFC 3310 [6], the following apply:

a) In the first request from the UE to the BSF, the UE shall include the private user identity IMPI in the "username" parameter of the Authorization header of HTTP request.

b) In the message from the BSF to the UE, the BSF shall include bootstrapping transaction identifier (B-TID) and the key lifetime to an XML document in the HTTP response payload. The BSF may also include additional server specific data to the XML document. The XML schema definition of this XML document is given in Annex C.

c) Authentication-Info header shall be included into the subsequent HTTP response after the BSF concluded that the UE has been authenticated. Authentication-Info header shall include the "rspauth" parameter.

After successful bootstrapping procedure the UE and the BSF shall contain the key material (Ks) and the ~~transaction identifier~~B-TID. The key material shall be derived from AKA parameters as specified in 3GPP TS 33.220 [1]. In addition, BSF shall also contain a set of security specific attributes related to the UE.

An example flow of successful bootstrapping procedure can be found in clause A.3.

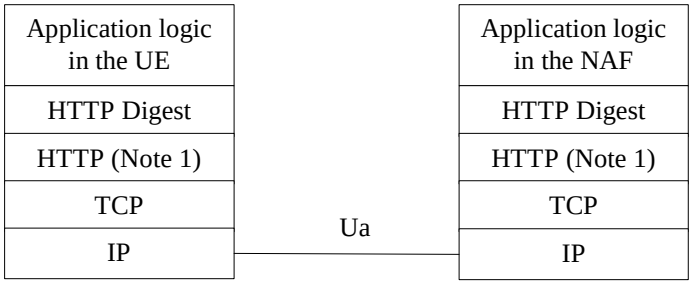
----------NEXT CHANGE----------

5.2.1 General

The HTTP Digest authentication model as described in RFC 2617 [9] can be used with bootstrapped security association as the authentication and integrity protection method, if the application protocol used over Ua interface between UE and NAF is based on HTTP. The HTTP Digest authentication may be used for all protocols that have adopted the HTTP authentication framework to mutually authenticate the UE and the NAF, and also optionally integrity protect any payload being transferred between them.

The UE shall indicate to an application server (i.e. a NAF) that it supports 3GPP-bootstrapping based HTTP Digest authentication by including a "product" token to the "User-Agent" header (cf. RFC 2616 [14]) that is a static string "3gpp-gba", which identifies the feature, i.e. support of GBA-based authentication. The User-Agent header field with this "product" token shall be added to each outgoing HTTP request if the UE supports GBA-based authentication using HTTP Digest. Upon receiving this "product" token, the application server if it supports NAF functionality may decide to authenticate the UE using GBA-based shared secret by executing the authentication procedure.

The protocol stack of the Ua interface when HTTP Digest authentication is used is presented in figure 5.2-1. The details are defined in the following subclauses.



NOTE 1: HTTP is not the only protocol that can be used. Other protocols can also be used as long as the protocol has adopted the HTTP authentication framework.

Figure 5.2-1: Protocol stack of Ua interface with HTTP Digest authentication

~~Note 1: HTTP is not the only protocol that can be used. Other protocols can also be used as long as the protocol has adopted the HTTP authentication framework.~~

----------NEXT CHANGE----------

5.3.1 General

Prior to establishing HTTP, the UE and the NAF may perform authentication. Three different authentication mechanisms may be used for UE and NAF authentication:

a) Shared key-based UE authentication (HTTP Digest) with certificate-based NAF authentication (TLS);

b) Shared key-based mutual authentication between UE and NAF (PSK TLS), and;

c) Certificate based mutual authentication between UE and AS;

The protocol stack of the Ua interface when TLS is used is presented in figure 5.3.1-1. and described in subclause 5.3.2. The HTTP Digest authentication is described in subclause 5.2.

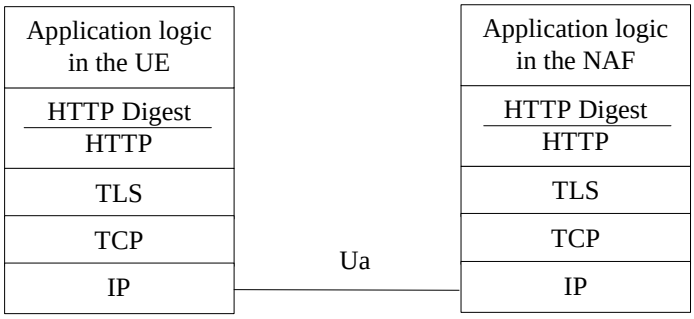


Figure 5.3.1-1: Protocol stack of Ua interface with TLS

The protocol stack of the Ua interface when PSK TLS is used is presented in figure 5.3.1-2 and described in subclause 5.3.3. The HTTP Digest authentication is described in subclause 5.2.

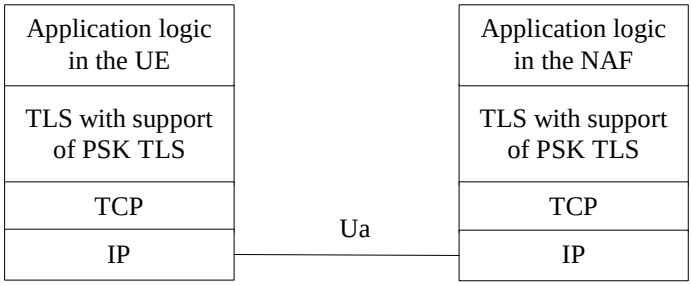


Figure 5.3.1-2: Protocol stack of Ua interface with PSK TLS~~5.3.2 Shared key-based UE authentication with certificate-based NAF authentication~~

5.3.2 Shared key-based UE authentication with certificate-based NAF authentication

5.3.2.1 Authentication procedure

The authentication mechanism described in this section is mandatory to implement in the UE and in the NAF.

The UE and the NAF shall support the TLS version as specified in RFC 2246 [11] and RFC 2818 [18]. See chapter 5.3.1 in TS 33.222 [5] for the detailed profiling of TLS.

a) When the UE starts communication via Ua reference point with the NAF, it shall establish a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate. The UE shall verify that the server certificate corresponds to the FQDN of the NAF it established the tunnel with. No client authentication is performed as part of TLS (no client certificate necessary).

b) The UE sends an HTTP request to the NAF inside the TLS tunnel (HTTPS, i.e. HTTP over TLS) as described in chapter 5.2.

c) The NAF shall authenticate the HTTP request using HTTP Digest as specified in subclause 5.2.

----------NEXT CHANGE----------

5.3.3.1 Authentication procedure

The authentication mechanism described in this clause is optional to implement in the UE and in the NAF.

The Pre-Shared Key Ciphersuites for TLS (PSK TLS) (draft-ietf-tls-psk-01 [15]) can be used with bootstrapped security association as the authentication, confidentiality, and integrity protection method.

The PSK TLS (draft-ietf-tls-psk-01 [15]) handshake shall be used with bootstrapped security association as follows:

- the ClientHello message shall contain one or more PSK-based ciphersuites;
- the ClientHello message shall contain the server_name TLS extension as specified in RFC 3546 [18] and it shall contain the hostname of the NAF;
- the ServerHello message shall contain a PSK-based ciphersuite selected by the NAF;
- the ServerKeyExchange shall contain the psk_identity_hint field and it shall contain a static string "3GPP-bootstrapping". The psk_identity_hint field may contain a list of psk_identity_hints (see NOTE 1);

NOTE 1: Other psk identity name spaces than "3GPP-bootstrapping" can be supported, however, they are out of the scope of this specification.~~;~~

- the ClientKeyExchange shall contain the psk_identity field and it shall contain a prefix "3GPP-bootstrapping"indicating selected psk identity name space, a separator character ";" and the B-TID;

- the UE shall derive the TLS premaster secret from the NAF specific key material (Ks_NAF) in the case of GBA_ME, and the NAF specific external key material (Ks_ext_NAF) in the case GBA_U as specified in draft-ietf-tls-psk-01 [15];

NOTE 2: The NAF specific internal key material (Ks_int_NAF) in the case of GBA_U shall not be used with PSK TLS.

An example flow of the PSK TLS procedure can be found in clause F.3.

----------NEXT CHANGE----------

A.3 Signalling flows demonstrating a successful bootstrapping procedure

The overall bootstrapping procedure in successful case is presented in figure A.3-1. The bootstrapping Zh interface performs the retrieval of an authentication vector by BSF from the HSS. The procedure corresponds to the step 2 in figure A.3-1.

This clause 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 later the bootstrapping key material generation procedure.

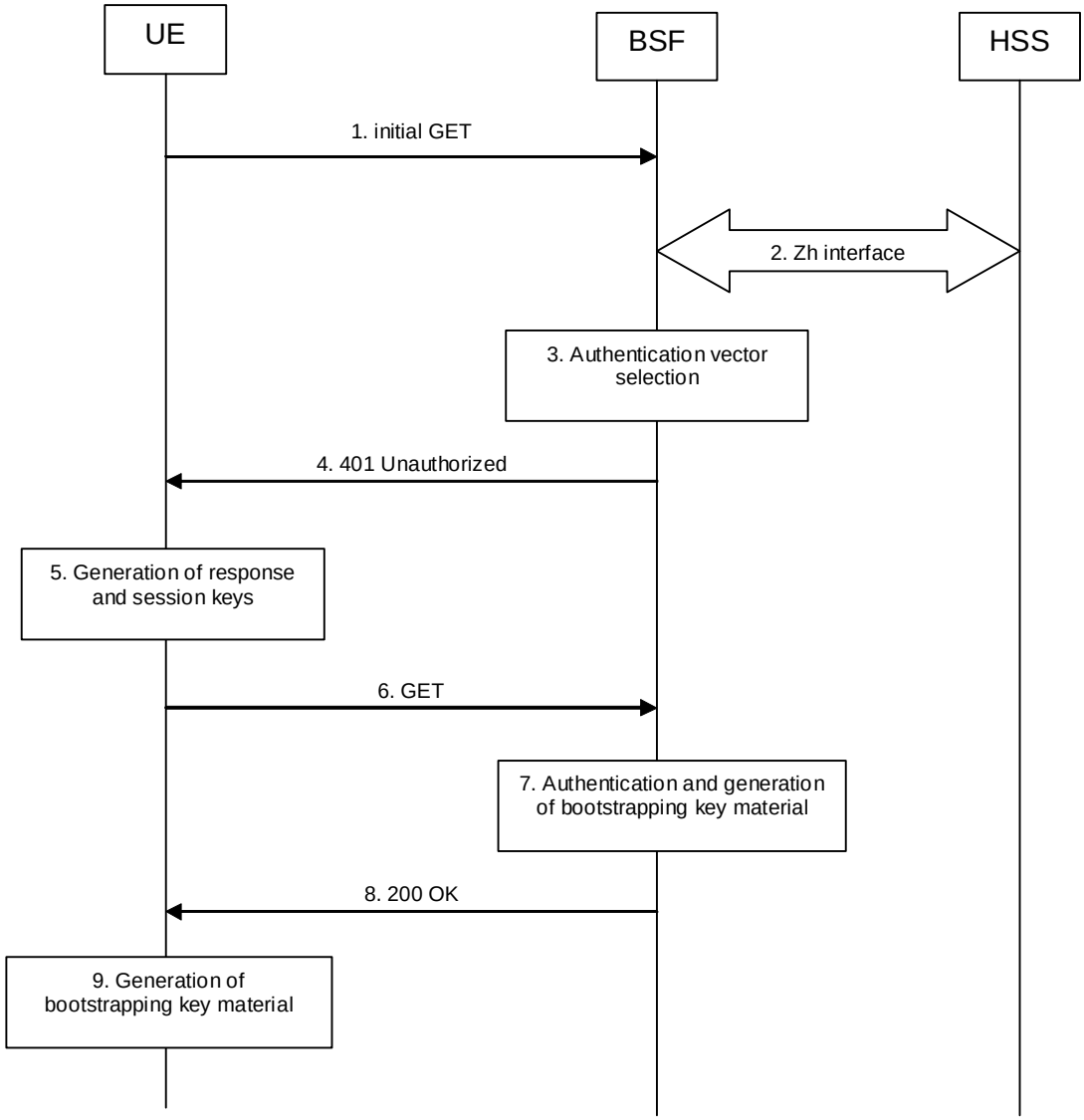


Figure A.3-1: Bootstrapping signalling

1. **Initial GET request (UE to BSF) - see example in table A.3-1**

   The purpose of this message is to initiate bootstrapping procedure between the UE and BSF. The UE sends an HTTP request containing the private user identity towards its home BSF.

Table A.3-1: Initial GET request (UE to BSF)

```
GET / HTTP/1.1
Host: registrar.home1.net:9999
User-Agent: Bootstrapping Client Agent; Release-6
Date: Thu, 08 Jan 2004 10:13:17 GMT
Accept: */*
Referer: http://pki-portal.home1.net:2311/pkip/enroll
Authorization: Digest username="user1_private@home1.net", realm="registrar.home1.net", nonce="",
uri="/", response=""
```

Request-URI: The Request-URI (the URI that follows the method name, "GET", in the first line) indicates the resource indication of this GET request. For bootstrapping server, this is by default "/".

Host: Specifies the Internet host and port number of the BSF server, obtained from the original URI given by referring resource.

User-Agent: Contains information about the user agent originating the request.

Date: Represents the date and time at which the message was originated.

Accept: Media types which are acceptable for the response.

Referer: Allows the user agent to specify the address (URI) of the resource from which the bootstrapping procedure was initiated.

Authorization: It carries authentication information. The private user identity (user1_private@home1.net) is carried in the username field of the Digest AKA protocol. The "uri" parameter (directive) contains the same value as the Request-URI. The ~~realm~~"realm" parameter (directive) contains the network name where the username is authenticated. The Request-URI and the "realm" parameter (directive) value are obtained from the same field in the USIM and therefore, are identical. In this example, it is assumed that a new UICC card was just inserted into the terminal, and there is no other cached information to send. Therefore, "nonce" and "response" parameters (directives) are empty.

2. **Zh: Authentication procedure**

BSF retrieves the corresponding AVs from the HSS.

For detailed signalling flows see 3GPP TS 29.109 [3].

Table A.3-2: BSF authentication information procedure (BSF to HSS)

Message source and destination	Zh Information element name	Information Source in GET	Description
BSF to HSS	Private User Identity	Authorization:	The Private User Identity is encoded in the username field according to the Authorization protocol.

3. **Authentication vector selection**

The BSF selects an authentication vector for use in the authentication challenge. For detailed description of the authentication vector, see 3GPP TS 33.203 [21].

NOTE 1: The authentication vector can be of the form as in 3GPP TS 33.203 [21] (if IMS AKA is the selected authentication scheme):

- $AV = RAND_n||AUTN_n||XRES_n||CK_n||IK_n$ where:
  - RAND: random number used to generate the XRES, CK, IK, and part of the AUTN. It is also used to generate the RES at the UE.
  - AUTN: Authentication token (including MAC and SQN); 128 bit value generated by the HSS.
  - XRES: Expected (correct) result from the UE.
  - CK: Cipher key (optional).
  - IK: Integrity key.

4. **401 Unauthorized response (BSF to UE) - see example in table A.3-3**

BSF forwards the challenge to the UE in HTTP 401 Unauthorized response (without the CK, IK and XRES). This is to demand the UE to authenticate itself. The challenge contains RAND and AUTN that are populated in nonce field according to RFC 3310 [6].

Table A.3-3: 401 Unauthorized response (BSF to UE)

```
HTTP/1.1 401 Unauthorized
Server: Bootstrapping Server; Release-6
Date: Thu, 08 Jan 2004 10:13:17 GMT
WWW-Authenticate: Digest realm="registrar.home1.net", nonce= base64(RAND + AUTN + server specific
data), algorithm=AKAv1-MD5, qop="auth-int"
```

Server: Contains information about the software used by the origin server (BSF).

Date: Represents the date and time at which the message was originated.

WWW-Authenticate: The BSF challenges the user. The nonce includes the quoted string, base64 encoded value of the concatenation of the AKA RAND, AKA AUTN and server specific data.

NOTE 2: The actual nonce value in the WWW-Authenticate header field is encoded in base64, and it can look like: nonce="A34Cm+Fva37UYWpGNB34JP".

5. **Generation of response and session keys at UE**

   Upon receiving the Unauthorized response, the UE extracts the MAC and the SQN from the AUTN. The UE calculates the XMAC and checks that XMAC matches the received MAC and that the SQN is in the correct range. If both these checks are successful the UE calculates the authentication challenge response (using RES and other parameters as defined in RFC 3310 [6]), and also computes the session keys IK and CK. The authentication challenge response is put into the Authorization header and sent back to the BSF in the GET request.

6. **GET request (UE to BSF) - see example in table A.3-4**

   The UE sends an HTTP GET request again, with the RES, which is used for response calculation, to the BSF.

Table A.3-4: GET request (UE to BSF)

```
GET / HTTP/1.1
Host: registrar.home1.net:9999
User-Agent: Bootstrapping Client Agent; Release-6
Date: Thu, 08 Jan 2004 10:13:18 GMT
Accept: */*
Referer: http://pki-portal.home1.net:2311/pkip/enroll
Authorization: Digest username="user1_private@home1.net", realm="registrar.home1.net",
nonce=base64(RAND + AUTN + server specific data), uri="/", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=AKAv1-MD5
```

Authorization: This carries the response to the authentication challenge received in step 4 along with the private user identity, the realm, the nonce, the URI, the qop, the NC, the cnonce, the response, the opaque, and the algorithm.

7. **Authentication and generation of key material at BSF**

   Upon receiving an integrity protected GET request carrying the authentication challenge response, the BSF checks that the expected response (calculated by the BSF using XRES and other parameter as defined in RFC 3310 [6]) matches the received challenge response. If the check is successful then the user has been authenticated and the private user identity is registered in the BSF.

   The BSF generates the bootstrapping transaction identifier (B-TID) for the IMPI and stores the tuple <B-TID,IMPI,CK,IK>.

   For detailed bootstrapping key material generation procedure see 3GPP TS 33.220 [1].

8. **200 OK response (BSF to UE) - see example in table A.3-5**

   The BSF sends 200 OK response to the UE to indicate the success of the authentication.

Table A.3-5: 200 OK response (BSF to UE)

```
HTTP/1.1 200 OK
Server: Bootstrapping Server; Release-6
Authentication-Info: qop=auth-int, rspauth="6629fae49394a05397450978507c4ef1",
cnonce="6629fae49393a05397450978507c4ef1", nc=00000001
Date:
Expires: Thu, 08 Jan 2004 10:23:17 GMT
Content-Type: application/vnd.3gpp.bsf+xml
Content-Length: (...)

<?xml version="1.0" encoding="UTF-8"?>
<BootstrappingInfo xmlns="uri:3gpp-gba">
  <btid>user@bsf.operator.com</btid>
```

```
   <lifetime>2004-05-28T13:20:00-05:00</lifetime>
</BootstrappingInfo>
```

Content-Type: Contains the media type of the entity body.

Content-Length: Indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.

Authentication-Info: This carries the server authentication information. The header includes the "rspauth" parameter which is calculated as specified in RFC 2617 [9] using RES for response calculation as specified in RFC 3310 [6].

Expires: Gives the date/time after which the response is considered stale.

9. **Generation of key material at UE**

The key material Ks is generated in UE by concatenating CK and IK. The NAF specific key material Ks_NAF is derived from Ks in the case of GBA_ME, or Ks_ext_NAF is derived from Ks_ext in the case of GBA_U, and used for securing the Ua interface. The UE stores the tuple <B-TID,Ks_NAF> or <B-TID,Ks_ext_NAF>.

For detailed bootstrapping key material generation procedure for NAF specific key (Ks_NAF or Ks_ext_NAF) see 3GPP TS 33.220 [1].

----------NEXT CHANGE----------

B.3 Signalling flows demonstrating a successful authentication procedure

The signalling flow in figure B.3-1 describes the generic message exchange between UE and NAF using HTTP Digest Authentication. The conversation can take place inside a server-authenticated TLS (as described in RFC 2246 [11]) tunnel in which case TLS session has been established before step 1.

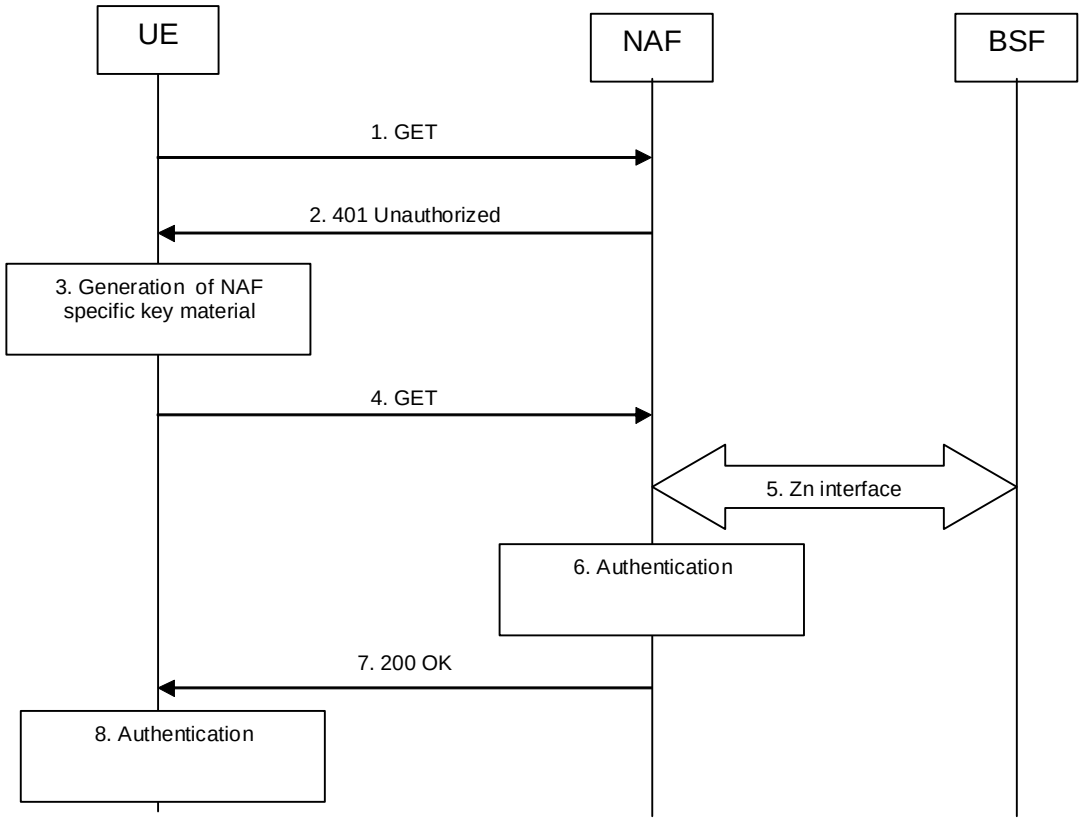


Figure B.3-1: HTTP Digest Authentication with bootstrapped security association

1. **GET request (UE to NAF) - see example in table B.3-1**

The UE sends an HTTP request to a NAF to gain access to a service.

Table B.3-1: Initial GET request (UE to BSF)

```
GET / HTTP/1.1
Host: naf1.home1.net:1234
User-Agent: NAF1 Application Agent; Release-6 3gpp-gba
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referrer: http://naf1.home1.net:1234/service
```

Request-URI: The Request-URI (the URI that follows the method name, "GET", in the first line) indicates the resource indication of this GET request.

Host: Specifies the Internet host and port number of the NAF server, obtained from the original URI given by referring resource.

User-Agent: Contains information about the user agent originating the request and it includes the static string "3gpp-gba" to indicate to the application server (i.e. NAF) that the UE supports 3GPP-bootstrapping based authentication.

Date: Represents the date and time at which the message was originated.

Accept: Media types which are acceptable for the response.

Referer: Allows the user agent to specify the address (URI) of the resource from which the URI for the NAF was obtained.

NOTE 1: This step can also be a POST request in which case the request would contain a client payload in the HTTP request and the corresponding Content-Type and Content-Length header values.

2. **401 Unauthorized response (NAF to UE) - see example in table B.3-2**

Upon receiving an HTTP request that contains static string "3gpp-gba" in the User-Agent header, NAF can choose to authenticate the UE using bootstrapped security association. If NAF chooses to authenticate the UE using bootstrapped security association, it responds with HTTP response code 401 "Unauthorized" which contains a WWW-Authenticate header. The header instructs the UE to use HTTP Digest Authentication with a bootstrapped security association.

Table B.3-2: 401 Unauthorized response (NAF to UE)

```
HTTP/1.1 401 Unauthorized
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
WWW-Authenticate: Digest realm="3GPP-bootstrapping@naf.home1.net",
nonce="6629fae49393a05397450978507c4ef1", algorithm=MD5, qop="auth,auth-int",
opaque="5ccc069c403ebaf9f0171e9517f30e41"
```

Server: Contains information about the software used by the origin server (NAF).

Date: Represents the date and time at which the message was originated.

WWW-Authenticate: The NAF challenges the user. The header instructs the UE to use HTTP Digest Authentication with a bootstrapped security association.

The options for the quality of protection (qop) attribute is by default "auth-int" meaning that the payload of the following HTTP requests and responses should integrity protected. If the conversation is taking place inside a server-authenticated TLS tunnel, the options for the qop attribute can also contain "auth" meaning that the payload of the following HTTP requests and responses are not protected by HTTP Digest. The integrity protection is handled on the TLS layer instead.

The realm attribute contains two parts delimited by "@" sign. The first part is a constant string "3GPP-bootstrapping" instructing the UE to use a bootstrapped security association. The second part is the FQDN of the NAF.

3. **Generation of NAF specific keys at UE**

UE verifies that the second part of the realm attribute does correspond to the server it is talking to. In particular, if the conversation is taking place inside a server-authenticated TLS tunnel, the UE shall verify that the server name in the server's TLS certificate matches the server name in the realm attribute of the WWW-Authenticate header.

UE derives the NAF specific key material Ks_NAF as specified in 3GPP TS 33.220 [1].

NOTE 2: If UE does not have a bootstrapped security association available, it obtains one by running bootstrapping procedure over Ub interface

4. **GET request (UE to NAF) - see example in table B.3-3**

UE generates the HTTP request by calculating the Authorization header values using the bootstrapping transaction identifier B-TID it received from the BSF as the username and the NAF specific key material Ks_NAF (base64 encoded) as the password, and sends the request to NAF.

Table B.3-3: GET request (UE to NAF)

```
GET / HTTP/1.1
Host: naf1.home1.net:1234
User-Agent: NAF1 Application Agent; Release-6 3gpp-gba
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://naf1.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@naf.home1.net",
nonce="a6332ffd2d234==", uri="/", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=MD5
```

Authorization: This carries the response to the authentication challenge received in step 2 along with the username, the realm, the nonce, the URI, the qop, the NC, the cnonce, the response, the opaque, and the algorithm.

The qop attribute is set to "auth-int" by default. If the conversation is taking place inside a server-authenticated TLS tunnel, the qop attribute can be set to "auth" as well.

NOTE 3: If step 1 was a POST request then this request would also be POST request and contain the same client payload in the HTTP request as was carried in step 1.

5. **Zn: NAF specific key procedure**

NAF retrieves the NAF specific key material (Ks_NAF) from the BSF.

For detailed signalling flows see 3GPP TS 29.109 [3].

Table B.3-4: Bootstrapping authentication information procedure (NAF to BSF)

Message source and destination	Zn Information element name	Information Source in GET	Description
NAF to BSF	B-TID	Authorization	The bootstrapping transaction identifier is encoded in the username field according to the Authorization protocol.

6. **Authentication at NAF**

NAF verifies the Authorization header by using the bootstrapping transaction identifier B-TID and the key material Ks_NAF obtained from BSF. NAF calculates the corresponding digest values using Ks_NAF, and compares the calculated values with the received values in the Authorization header.

The NAF also verifies that the DNS name in the realm attribute matches its own. If the conversation is taking place inside a server-authenticated TLS tunnel, the NAF shall also verify that this DNS name is the same as that of the TLS server certificate.

If the verification succeeds, the incoming client-payload request is taken in for further processing.

7. **200 OK response (NAF to UE) - see example in table B.3-5**

The NAF sends 200 OK response to the UE to indicate the success of the authentication. NAF generates a HTTP response containing the server-payload it wants to send back to the UE. The NAF can use key material Ks_NAF to integrity protect and authenticate the response.

Table B.3-5: 200 OK response (NAF to UE)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27Content-Type: text/html
Content-Length: 1234
Authentication-Info: qop=auth-int, rspauth="6629fae49394a05397450978507c4ef1",
cnonce="6629fae49393a05397450978507c4ef1", nc=00000001
Date: Thu, 08 Jan 2004 10:50:35 GMT
Expires: Fri, 09 Jan 2004 10:50:36 GMT

<SERVER PAYLOAD>
```

Content-Type: Contains the media type of the entity body.

Content-Length: Indicates the size of the entity-body, in decimal number of OCTETs, sent to the recipient.

Authentication-Info: This carries the protection

Expires: Gives the date/time after which the response is considered stale.

8. **Authentication at UE**

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

NOTE 4: Additional messages can be exchanged using steps 4 through 8 as many times as is necessary. The following HTTP ~~request~~requests and responses are constructed according to RFC 2617 [9].

----------NEXT CHANGE----------

C.1 Introduction

This annex contains the XML schema definition for an XML document carrying the bootstrapping transaction identifier (B-TID), the key lifetime, and possibly other server specific data.

The "lifetime" attribute shall indicate the expiry time of the key.

~~Editor's note: The content-type "application/vnd.3gpp.bsf+xml" needs to be registered with IANA.~~

Editor's note: The content-type "application/vnd.3gpp.bsf+xml" needs to be registered with IANA.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="uri:3gpp-gba"
    xmlns:gba="uri:3gpp-gba"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!-- definition of the root element containing B-TID and key lifetime -->
  <xs:complexType name="bootstrappingInfoType">
    <xs:sequence>
       <xs:element name="btid" type="xs:string"/>
       <xs:element name="lifetime" type="xs:dateTime"/>
    </xs:sequence>
  </xs:complexType>

  <!-- the root element -->
  <xs:element name="BootstrappingInfo" type="gba:bootstrappingInfoType"/>

</xs:schema>
```