3GPP TSG-CN Meeting #25 **NP-040366**
8th – 10th September 2004. Palm Springs, USA.

Source:	**TSG CN WG 1**
Title:	**TS 24.109 v.2.1.1, Bootstrapping interface (Ub) and Network application function interface (Ua); Protocol details**
Agenda item:	**9.22**
Document for:	**APPROVAL**

Presentation of Technical Specification to TSG

Presentation to:	**TSG CN Meeting #25**
Document for presentation:	**TS 24.109, version 2.1.1**
Presented for:	**Approval**

Abstract of document:
TS 24.109 describes stage 3 for the HTTP Digest AKA based implementation of Ub interface (UE-BSF) and the HTTP Digest and the PSK TLS based implementation of bootstrapped security association usage over Ua interface (UE-NAF) in Generic Authentication Architecture (GAA) as specified in 3GPP TS 33.220. The purpose of the Ub interface is to create a security association between UE and BSF for further usage in GAA applications. The purpose of the Ua interface is to use the so created bootstrapped security association between UE and NAF for secure communication.
The present document also defines stage 3 for the Authentication Proxy usage as specified in 3GPP TS 33.222.
The present document also defines stage 3 for the subscriber certificate enrolment as specified in 3GPP TS 33.221 [4] which is one realization of the Ua interface. The subscriber certificate enrolment uses the HTTP Digest based implementation of bootstrapped security association usage to enrol a subscriber certificate and the delivery of a CA certificate.

Changes since last presentation to TSG Meeting:
None. The TS is presented first time to the plenary.

Outstanding Issues:
Remaining topics:
- more detailed description of Authentication Proxy usage;
- further enhancement of the XML schema that defines the XML document carrying the bootstrapping key lifetime, and possibly other server specific data from the BSF to the UE.

Contentious Issues:
None.

3GPP TS 24.109 V2.0.0 (2004-09)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Core Network; Bootstrapping interface (Ub) and Network application function interface (Ua); Protocol details (Release 6)

Keywords
UMTS, Network, IP, SIP, SDP, multimedia

3GPP

Postal address

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

Internet
http://www.3gpp.org

Contents

Foreword

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

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

Version x.y.z

where:

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

1 Scope

The present document defines stage 3 for the HTTP Digest AKA [5] based implementation of Ub interface (UE-BSF) and the HTTP Digest [8] and the PSK TLS [14A] based implementation of bootstrapped security association usage over Ua interface (UE-NAF) in Generic Authentication Architecture (GAA) as specified in 3GPP TS 33.220 [1]. The purpose of the Ub interface is to create a security association between UE and BSF for further usage in GAA applications. The purpose of the Ua interface is to use the so created bootstrapped security association between UE and NAF for secure communication.

The present document also defines stage 3 for the Authentication Proxy usage as specified in 3GPP TS 33.222 [4A].

The present document also defines stage 3 for the subscriber certificate enrolment as specified in 3GPP TS 33.221 [4] which is one realization of the Ua interface. The subscriber certificate enrolment uses the HTTP Digest based implementation of bootstrapped security association usage to enrol a subscriber certificate and the delivery of a CA certificate.

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".

[4A] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

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

[6] 3GPP TS 23.003: "Numbering, addressing and identification".

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

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

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

[10] 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".

[14A] IETF draft-ietf-tls-psk-01: "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)".

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

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

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

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

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

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

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

[19] Open Mobile Alliance: "WPKI"

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

[20] 3GPP TS 33.203: "3G security; Access security for IP-based services".

3 Definitions, symbols 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: signs all certificates that it issues with its private key
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

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 [18], and OMA WPKI [19] specifications).

3.2 Symbols

Void.

3.3 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
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	WAP Identity Module
WPKI	Wireless PKI
WTLS	Wireless Transport Layer Security
XRES	Expected authentication response

4 Generic Bootstrapping Architecture; Ub interface

4.1 Introduction

Generic Authentication Architecture (GAA) is based on shared secrets provided by generic bootstrapping architecture (GBA). The stage 2 description of GAA framework is described in 3GPP TR 33.919 [2] and the GBA procedures in 3GPP TS 33.220 [1].

The GBA related to the Ub interface is between the UE and bootstrapping server function (BSF). During the bootstrapping procedure BSF also uses the Zh interface to request authentication vectors from HSS. The Zh interface is defined in 3GPP TS 29.109 [3]. The end result of the bootstrapping procedure is that both BSF and an UE have a security association in a form of a bootstrapping transaction identifier (B-TID) and key material (Ks).

According to 3GPP TS 33.220 [1] the bootstrapping procedure shall be based on HTTP Digest AKA as described in RFC 3310 [5]. The protocol stack of the Ub interface in bootstrapping procedure is presented in figure 4.1-1. The details are defined in the following subclauses.

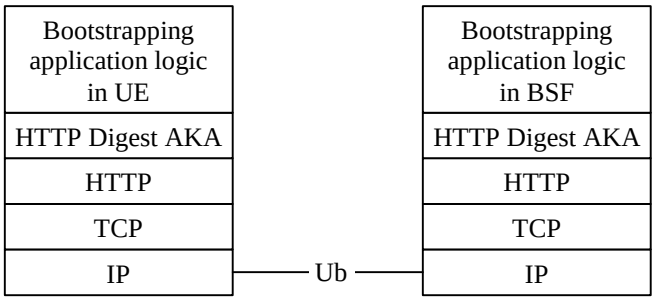

Figure 4.1-1: Protocol stack of Ub interface

The bootstrapping procedure described in the present document can result to different key materials depending on whether ME-based or UICC-based GBA is used. However, the bootstrapping procedure itself is the same for both ME-based GBA (GBA_ME), and UICC-based GBA (GBA_U).

4.2 Bootstrapping procedure

A UE and the BSF shall establish bootstrapped security association between them by running bootstrapping procedure. Bootstrapping security association consists of a transaction identifier 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 [5] with the modifications described below.

In the bootstrapping procedure, Authorization, WWW-Authenticate, and Authentication-Info HTTP headers shall be used as described in RFC 3310 [5] with following exceptions:

- the "realm" parameter shall contain the network name where the username is authenticated;
- the quality of protection ("qop") parameter shall be "auth-int"; and
- the "username" parameter shall contain user's private identity (IMPI).

NOTE: If the UE does not have an IMS subscription, the IMPI will be constructed from IMSI, according to 3GPP TS 23.003 [6].

In addition to RFC 3310 [5], the following shall be added:

1. 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.
2. In the message from the BSF to the UE, the BSF shall include bootstrapping 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.

Editor's note: It is FFS whether the B-TID will be included in the message from the BSF to the UE.

3. 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. 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.

4.3 User authentication failure

If the response returned by the UE is different than expected, the BSF may challenge the UE again with a new AKA challenge. After N consecutive incorrect responses from the UE, the BSF shall indicate a failure to the UE. The exact value of N is defined by local policy.

4.4 Network authentication failure

In case the UE fails at authenticating the network, the UE shall abort the bootstrapping procedure.

4.5 Synchronization failure

If the UE considers the sequence number in the challenge not to be in the correct, the UE shall send a synchronization failure indication back to the BSF as specified RFC 3310 [5].

An example flow can be found in clause A.4.

5 Network application function; Ua interface

5.1 Introduction

The usage of bootstrapped security association, i.e. B-TID and Ks_NAF (or Ks_ext_NAF) over Ua interface depends on the application protocol used between UE and NAF (e.g. a PKI portal, see 3GPP TS 33.221 [4]).

The Ua interface is used to supply the B-TID, generated during the bootstrapping procedure, to the network application function (NAF), and Zn interface is used by the NAF to retrieve the Ks_NAF or Ks_ext_NAF from BSF. The Ua interface depends on type of NAF. The Zn interface is defined in 3GPP TS 29.109 [3]. This clause describes how B-TID and Ks_NAF or Ks_ext_NAF can be utilized in HTTP Digest authentication as described in RFC 2617 [8].

5.2 HTTP Digest

The HTTP Digest authentication model as described in RFC 2617 [8] 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 described in subclause 5.2.1.

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.

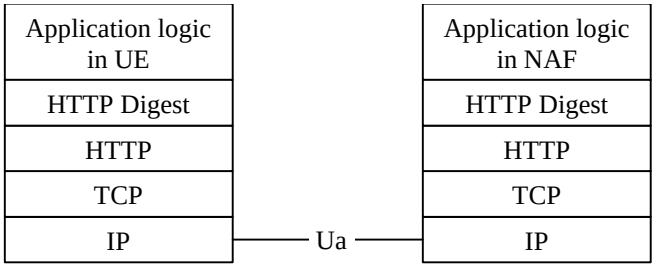

Figure 5.2-1: Protocol stack of Ua interface with HTTP Digest authentication

5.2.1 Authentication procedure

5.2.1.1 General

HTTP Digest authentication [8] shall be used with previously bootstrapped security association as follows:

- the "username" parameter shall be the bootstrapping transaction identifier;
- the password used in the digest calculations shall be 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 of GBA_U The NAF specific key material (Ks_NAF or Ks_ext_NAF) is Base64 encoded as specified in RFC 3548 [9]; and

NOTE 1: The NAF specific key material (Ks_NAF or Ks_ext_NAF) is derived from the key material (Ks or Ks_ext) using key derivation function as specified in 3GPP TS 33.220 [1].

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

- the "realm" parameter shall contain two parts, first part shall text string "3GPP-bootstrapping@", and the latter part shall be the FQDN of the NAF (e.g. "3GPP-bootstrapping@naf1.operator.com").

Both the UE and the NAF shall verify upon receiving each of the HTTP responses and HTTP requests that the second part of the realm attribute is equal to the FQDN of the NAF.

5.2.1.2 Integrity protection

Integrity protection may be provided:

- by using HTTP Digest integrity protection, i.e. quality of protection (qop) parameter is set to "auth-int"; or
- by using server-authenticated TLS tunnel as described in RFC 2818 [12].

If server-authenticated TLS tunnel is used, both UE and NAF shall verify upon receiving each of the HTTP responses and HTTP requests that the second part of the realm attribute is equal to FQDN included in the server's certificate.

5.2.2 Authentication failures

Authentication failures are handled as they are described in RFC 2617 [8].

5.2.3 Bootstrapping required indication

NAF shall indicate to the UE that bootstrapped security association is required by sending an HTTP response with code 401 "Unauthorized" and include the WWW-Authenticate header into the response. In particular, the "realm" attribute shall contain a prefix "3GPP-bootstrapping@" and this shall trigger UE to run bootstrapping procedure over Ub interface.

5.2.4 Bootstrapping renegotiation indication

The NAF shall indicate to the UE that the existing bootstrapped security association used in the last HTTP request sent by the UE has expired and that a new bootstrapped security association is required by sending an HTTP response described in subclause 5.2.3. When the UE receives the 401 "Unauthorized" HTTP response to the HTTP request that was protected using the existing bootstrapped security association, this shall trigger the UE to run bootstrapping procedure over Ub interface.

5.3 Pre-shared key TLS

5.3.1 Authentication procedure

The Pre-Shared Key Ciphersuites for TLS (PSK TLS) (draft-ietf-tls-psk-01 [14A]) can be used with bootstrapped security association as the authentication, confidentiality, and integrity protection method.

The protocol stack of the Ua interface when PSK TLS authentication is used is presented in figure 5.3-1.

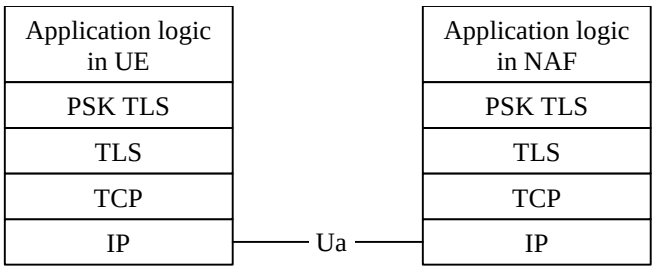

Figure 5.3-1: Protocol stack of Ua interface with PSK TLS

The PSK TLS (draft-ietf-tls-psk-01 [14A]) 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 [17] 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 ClientKeyExchange shall contain the psk_identity field and it shall contain 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 [14A];

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

5.3.2 Authentication failures

Authentication failures are handled as they are described in RFC 2246 [10] and in draft-ietf-tls-psk-01 [14A].

5.3.3 Bootstrapping required indication

During TLS handshake, the NAF shall indicate to the UE that bootstrapped security association is required by sending a ServerHello message containing a PSK-based ciphersuite, and a ServerKeyExchange message containing the psk_identity_hint field which contains a static string "3GPP-bootstrapping". This shall trigger the UE to run the bootstrapping procedure over Ub interface.

NOTE: The NAF shall select a PSK-based ciphersuite only if the UE has offered one or more PSK-based ciphersuites in the corresponding ClientHello message.

5.3.4 Bootstrapping renegotiation indication

During usage of TLS session, the NAF shall indicate to the UE that bootstrapped security association has expired by sending close_notify alert message to the UE. The UE may attempt resume the old TLS session by sending a ClientHello message containing the old session ID. The NAF shall refuse to use the old session ID by sending a ServerHello message with a new session ID. This will indicate to the UE that the bootstrapped security association it used has expired.

During TLS handshake, the NAF shall indicate to the UE that the bootstrapped security association has expired by sending handshake_failure message as a response to the Finished message sent by the UE. This will indicate to the UE that the bootstrapped security association it used has expired.

6 PKI portal, Ua interface

6.1 Introduction

3GPP TS 33.221 [4] specifies the enrolment of subscriber certificates and the delivery of CA certificates to the UE. The TS specifies that the authentication of these procedures be based on bootstrapping procedure and more generally on the HTTP Digest authentication as described in subclause 5.2 of the present document.

6.2 Subscriber certificate enrolment

The subscriber certificate enrolment procedure contains the following requests:

- an enrolment request in the form of PKCS#10 [15];
- an optional request for WIM specific authentication code for key generation [18]; and
- an optional request for WIM specific authentication code for proof of key origin [18].

Respectively, the subscriber certificate enrolment procedure contains the following responses:

- a subscriber certificate; and
- a WIM specific authentication code [18].

NOTE: The on board key generation and the generation of the proof of key origin requires a WIM specific authentication code. Whether it is, is decided by the issuer of the WIM card.

6.2.1 Enrolment procedure

The UE shall generate a PKCS#10 certification request [15] according to 3GPP TS 33.221 [4]. The UE shall send the PKCS#10 certification request to the PKI portal in the HTTP payload in a HTTP POST request. The Request-URI shall indicate the desired response type. Upon successful enrolment, PKI portal shall return the enrolled subscriber certificate in the desired format.

The UE populates the HTTP POST request as follows:

- the HTTP version shall be 1.1 which is specified in RFC 2616 [14];
- the base of the Request-URI is extracted from the full PKI portal URI (e.g. if the full PKI portal URI is "http://pki-portal.operator.com/enrol" then the Request-URI shall be "/enrol".

NOTE 1: In case a proxy is used between the UE and the PKI portal, then the full Request-URI will be used in the HTTP Post request.

- the Request-URI shall contain an URI parameter "response" that shall be set to "single", "pointer", or "chain" depending on the UE's desired response type (e.g. Request-URI may take the form of "/enrol?response=single" for certificate delivery);

NOTE 2: The PKI portal might ignore the UE's desired response type, and the UE should be capable of receiving the issued subscriber certificate in any of the response types.

- the UE may add additional URI parameters to the Request-URI;

NOTE 3: The PKI portal might ignore the additional URI parameters.

- the HTTP header Content-Type shall be "application/x-pkcs10";

- the HTTP header Content-Length shall be the length of the Base64 encoded PKCS#10 certification request in Octets; and

- the HTTP payload shall contain the Base64 encoded PKCS#10 certification request and optionally surrounded by "----- BEGIN CERTIFICATE REQUEST -----" and "----- END CERTIFICATE REQUEST -----" tags;

- the UE may add additional HTTP headers to the HTTP POST request.

The UE sends the HTTP POST request to the PKI portal. The PKI portal checks that the HTTP request is valid, and extracts the Base64 encoded PKCS#10 certification request for further processing. The PKI portal shall verify that the subscriber is authorized to receive the particular type of certificate by checking subscriber's user security settings received from the BSF as specified 3GPP TS 33.220 [4].

Upon successful subscriber certificate creation procedure, the PKI portal shall return the subscriber certificate to the UE in the UE's desired format or in the PKI portal's desired format.

The response format type shall be one of the following:

- the subscriber certificate itself (i.e. desired response type was "single");

- a pointer to the subscriber certificate (i.e. desired response type was "pointer"); or

- a certificate chain that contains full certification chain from subscriber certificate to the root certificate (i.e. desired response type was "chain").

If response format type is "single", the PKI portal shall populate HTTP response as follows:

- the HTTP status code shall be 200;

- the HTTP header Content-Type shall be "application/x-x509-user-cert";

- the HTTP header Content-Length shall be the length of the HTTP payload in octets;

- the HTTP payload shall contain the Base64 encoded subscriber certificate and optionally surrounded by "----- BEGIN CERTIFICATE -----" and "----- END CERTIFICATE -----" tags;

- the PKI portal may add additional HTTP headers to the HTTP response.

If response format type is "pointer", the PKI portal shall populate HTTP response as follows:

- the HTTP status code shall be 200;

- the HTTP header Content-Type shall be "application/vnd.wap.cert-response";

- the HTTP header Content-Length shall be the length of the HTTP payload in octets;

- the HTTP payload shall contain the Base64 encoded CertResponse structure and optionally surrounded by "----- BEGIN CERTIFICATE RESPONSE -----" and "----- END CERTIFICATE RESPONSE -----" tags;

- the PKI portal may add additional HTTP headers to the HTTP response.

If response format type is "chain", the PKI portal shall populate HTTP response as follows:

- the HTTP status code shall be 200;
- the HTTP header Content-Type shall be "application/pkix-pkipath";
- the HTTP header Content-Length shall be the length of the HTTP payload in octets;
- the HTTP payload shall contain the Base64 encoded PkiPath structure;
- the PKI portal may add additional HTTP headers to the HTTP response.

The PKI portal shall send the HTTP response to the UE. The UE shall check that the HTTP response is valid, and extract the Base64 encoded subscriber certificate, pointer to the subscriber certificate, or certificate chain for further processing.

An example flow of a successful subscriber certificate enrolment procedure can be found in clause E.3.

6.2.2 WIM specific authentication code for key generation

The UE may be equipped with a WIM which may require an authentication code from WIM provider in order to generate a key onboard WIM as specified in OMA ECMAScript [18] and OMA WPKI [19] specifications. In this case, the UE shall request the authentication code from PKI portal using a HTTP GET request. If the PKI portal can acquire authentication code, it is returned to the UE in the corresponding HTTP response.

The UE populates the HTTP GET request as follows:

- the HTTP version shall be 1.1 which is specified in RFC 2616 [14];
- the base of the Request-URI is extracted from the full PKI portal URI and appended with "/wim-auth-code" (e.g. if the full PKI portal URI is "http://pki-portal.operator.com/enrol" then the Request-URI shall be "/enrol/wim-auth-code");
- the Request-URI shall contain an URI parameter "request" that shall be set to the return value received from the WIM;

NOTE 1: If an authentication code is required, the WIM will return "error:AuthReq:cardSerialNumber:Challenge". The cardSerialNumber and the Challenge are in a hexadecimal format as specified in OMA ECMAScript specification [18].

- the UE may add additional URI parameters to the Request-URI;
- the UE may add additional HTTP headers to the HTTP GET request.

The UE sends the HTTP GET request to the PKI portal. The PKI portal acknowledges that this is an authentication code because the Request-URI contains the "/wim-auth-code" and the URI parameter "request". The PKI portal extracts the authentication code derivation parameters from the URI parameter "request", and derives the authentication code.

NOTE 2: The actual derivation of the authentication code is outside the scope.

Upon successful authentication code derivation, the PKI portal shall return the authentication code to the UE in a HTTP response:

- the HTTP status code shall be 200;
- the HTTP header Content-Type shall be "text/plain";
- the HTTP header Content-Length shall be the length of the HTTP payload in octets;
- the HTTP payload shall contain the authentication code in a hexadecimal format;
- the PKI portal may add additional HTTP headers to the HTTP response.

Upon receiving the authentication code from the PKI portal, the UE shall use it to authenticate the procedure of generating the key onboard the WIM.

6.2.3 WIM specific authentication code for proof of key origin

The UE may be equipped with a WIM which may require an authentication code from WIM provider in order to generate a proof of key origin onboard WIM as specified in OMA ECMAScript [18] and OMA WPKI [19] specifications. In this case, the UE shall request the authentication code from PKI portal using a HTTP GET request. If the PKI portal can acquire authentication code, it is returned to the UE in the corresponding HTTP response.

The procedure to obtain the authentication code for the generation of proof of key origin onboard WIM is the same as for the key generation, and is described in subclause D.2.2.

Upon receiving the authentication code from the PKI portal, the UE shall use it to authenticate the procedure generating the proof of key origin onboard the WIM.

6.2.4 Error situations

Subscriber certificate enrolment may not be successful for multiple reasons. The error cases are indicated by using 4xx and 5xx HTTP Status Codes as defined in RFC 2616 [14]. The 4xx status code indicates that the UE seems to have erred, and the 5xx status code indicates that the PKI portal is aware that it has erred. Possible error situations during subscriber certificate enrolment and their mappings to HTTP Status Codes are described in table 6.2.4-1.

NOTE: On the table 6.2.4-1, the "Description" column describes the error situation in PKI portal. The "PKI portal error" column describes the typical reason for the error.

An example flow of a failure in subscriber certificate enrolment procedure can be found in clause E.4.

Table 6.2.4-1: HTTP Status Codes used for enrolment error

HTTP Status Code	HTTP Error	UE should repeat the request	Description	PKI portal error
400	Bad Request	No	Request could not be understood	PKCS#10 request was missing, or malformed
401	Unauthorized	Yes	Request requires authentication (cf. subclause 5.2)	Authentication pending, cf. subclause 5.2
402	Payment Required	No	Reserved for future use	-
403	Forbidden	No	PKI portal understood the request, but is refusing to fulfil it	PKCS#10 request was valid, but subscriber is not allowed to enrol this particular type of certificates or PKCS#10 request contained unacceptable parameters
404	Not Found	No	PKI portal has not found anything matching the Request-URI	The Request-URI was malformed and PKI portal cannot fulfil the enrolment request
405 to 406	*	No	Not used by PKI portal	-
407	Proxy Authentication Required	Yes	PKI portal uses Authentication Proxy and UE shall authenticate itself with the proxy	Authentication Proxy authentication pending, cf. subclause 5.2
408 to 417	*	No	PKI portal should not use these status codes	-
500	Internal Server Error	No	PKI portal encountered an unexpected error	PKI portal is mis-configured
501	Not Implemented	No	PKI portal does not support the required functionality	The server does not contain PKI portal service
502	Bad Gateway	No	Gateway/Proxy received an invalid response from PKI portal	PKI portal is behind a gateway/proxy and sent an invalid response to the gateway/proxy
503	Service Unavailable	Yes	PKI portal service is currently unavailable	PKI portal is temporarily unavailable, UE may repeat the request after delay indicated by "Retry-After" header

HTTP Status Code	HTTP Error	UE should repeat the request	Description	PKI portal error
504	Gateway Timeout	No	Gateway/Proxy did not receive a timely response from the upstream server	PKI portal is behind a gateway/proxy and did not send a response to the gateway/proxy in time, or was not reachable by the gateway/proxy
505	HTTP Version Not Supported	No	PKI portal does not support the HTTP protocol version that was used in the request line	UE should use HTTP/1.1 version with PKI portal

6.3 CA certificate delivery

The root certificate delivery procedure contains the following request:

- a CA certificate delivery request;

and the corresponding response:

- the CA certificate.

6.3.1 CA certificate delivery procedure

The UE shall populate the HTTP GET request as follows:

- the HTTP version shall be 1.1 which is specified in RFC 2616 [14];
- the base of the Request-URI is extracted from the full PKI portal URI (e.g. if the full PKI portal URI is "http://pki-portal.operator.com/getcertificate" then the Request-URI shall be "/getcertificate".

NOTE 1: In case a proxy is used between the UE and the PKI portal, then the full Request-URI will be used in the HTTP Post request.

- the Request-URI shall contain an URI parameter "in" that shall be the Base64 encoding of the DER encoded Issuer field of the X.509 certificate;
- the Request-URI may contain an URI parameter "ki" that shall be the Base64 encoding of the DER encoded the Key Identifier of the X.509 certificate;

NOTE 2: Key Identifier of the CA certificate can be obtained from the Authority Key Identifier extension of the subscriber certificate.

- the UE may add additional URI parameters to the Request-URI;
- the UE may add additional HTTP headers to the HTTP GET request.

The UE sends the HTTP GET request to the PKI portal. The PKI portal checks that the HTTP request is valid, and extracts the "in" parameter and optionally "ki" parameter from the Request-URI. If the PKI portal can verify that the Issuer field parameter is valid, and that the UE may set the CA certificate as a root certificate (i.e. trusted CA certificate), it will then send the CA certificate back to the UE in the corresponding HTTP response.

The PKI portal shall populate the HTTP response as follows:

- the HTTP status code shall be 200;
- the HTTP header Content-Type shall be "application/x-x509-ca-cert";
- the HTTP header Content-Length shall be the length of the HTTP payload in octets;
- the HTTP payload shall contain the Base64 encoded CA certificate structure and optionally surrounded by "----- BEGIN CERTIFICATE -----" and "----- END CERTIFICATE -----" tags;
- the PKI portal may add additional HTTP headers to the HTTP response.

The PKI portal shall send the HTTP response to the UE. The UE shall check that the HTTP response is valid, and extract the Base64 encoded CA certificate for further processing. UE shall validate and match the received CA certificate against the parameters is supplied in the corresponding request.

An example flow of CA certificate procedure can be found in clause E.5.

6.3.2 Error situations

CA certificate delivery may not be successful for multiple reasons. The error cases are indicates by using 4xx and 5xx HTTP Status Codes as defined in RFC 2616 [14]. The 4xx status code indicates that the UE seems to have erred, and the 5xx status codes indicate that the PKI portal is aware that it has erred. Possible error situations during CA certificate delivery and their mappings to HTTP Status Codes are described in table 6.3.2-1.

NOTE: On the table 6.3.2-1, the "Description" column describes the error situation in PKI portal. The "PKI portal error" column describes the typical reason for the error.

An example flow of a failure in CA certificate delivery procedure can be found in clause E.6.

Table 6.3.2-1: HTTP Status Codes for CA certificate delivery error

HTTP Status Code	HTTP Error	UE should repeat the request	Description	PKI portal error
400	Bad Request	No	Request could not be understood	Request could not be understood
401	Unauthorized	Yes	Request requires authentication (cf. subclause 5.2)	Authentication pending, cf. subclause 5.2
402	Payment Required	No	Reserved for future use	-
403	Forbidden	No	PKI portal understood the request, but is refusing to fulfil it	CA certificate delivery request was understood but PKI portal refuses to deliver the CA certificate
404	Not Found	No	PKI portal has not found anything matching the Request-URI	PKI portal does not have the requested CA certificate
405 to 406	*	No	Not used by PKI portal	-
407	Proxy Authentication Required	Yes	PKI portal uses Authentication Proxy and UE shall authenticate itself with the proxy	Authentication Proxy authentication pending, cf. subclause 5.2
408 to 417	*	No	PKI portal should not use these status codes	-
500	Internal Server Error	No	PKI portal encountered an unexpected error	PKI portal is mis-configured
501	Not Implemented	No	PKI portal does not support the required functionality	The server does not contain PKI portal service
502	Bad Gateway	No	Gateway/Proxy received an invalid response from PKI portal	PKI portal is behind a gateway/proxy and sent an invalid response to the gateway/proxy
503	Service Unavailable	Yes	PKI portal service is currently unavailable	PKI portal is temporarily unavailable, UE may repeat the request after delay indicated by "Retry-After" header
504	Gateway Timeout	No	Gateway/Proxy did not receive a timely response from the upstream server	PKI portal is behind a gateway/(proxy and did not send a response to the gateway/proxy in time, or was not reachable by the gateway/proxy)
505	HTTP Version Not Supported	No	PKI portal does not support the HTTP protocol version that was used in the request line.	UE should use HTTP/1.1 version with PKI portal

7 Authentication Proxy

7.1 Introduction

The use of authentication proxy (AP) is specified in 3GPP TS 33.222 [4A]. The AP in GAA is used to separate the GAA specific authentication procedure and the Application Server (AS) specific application logic to different logical entities. The AP is configured as a HTTP reverse proxy, i.e. the FQDN of the AS is configured to the AP such a way that the IP traffic intended to the AS is directed to the AP by the network. The AP performs the GAA authentication of the UE. After the GAA authentication procedure has been successfully completed, the AP assumes the typical role of a reverse proxy, i.e. the AP forwards HTTP requests originating from the UE to the correct AS, and returns the corresponding HTTP responses from the AS to the originating UE.

7.2 Authentication

The authentication of the UE shall be based on GAA as specified in clause 5.

The UE may indicate the user identity intended to be used with the AS by adding a HTTP header to the outgoing HTTP requests. The HTTP header name shall be "X-3GPP-Intended-Identity" and it shall contain the user identity surrounded by quotation marks ("). If the HTTP header has been added, the AP shall verify that the user identity belongs to the subscriber.

7.3 Authorization

The AP shall be able to decide whether particular subscriber, i.e. the UE, is authorized to access a particular AS. The granularity of the authorization procedures is specified in 3GPP TS 33.222 [4A].

The AP may indicate an asserted identity or a list of identities to the AS by adding a HTTP header to the HTTP requests coming from the UE and forwarded to the AS. The HTTP header name shall be "X-3GPP-Asserted-Identity" and it shall contain a list of identities separated by comma (,) and each identity is surrounded by quotation marks ("). Whether the AP adds this HTTP header to the HTTP request depends on the subscriber's GBA user security settings.

Annex A (informative): Signalling flows of bootstrapping procedure

A.1 Scope of signalling flows

This annex gives examples of signalling flows for bootstrapping procedure.

A.2 Introduction

Editor's note: Material yet to be specified.

A.2.1 General

Bootstrapping procedure is executed in order to establish bootstrapped security association, i.e. bootstrapping session between an UE and the BSF.

The bootstrapping session is used between a UE and a NAF. An example usage of it is described in annex B.

A.2.2 Key required to interpret signalling flows

3GPP TS 24.228 [13], subclause 4.1.1, specifies the key required to interpret the contents of the SIP methods. This key is used with HTTP based messages (cf. RFC 2616 [14]) as well since SIP and HTTP messages resemble each other in structure. The following key rules are used in addition to those specified in 3GPP TS 24.228 [13]:

a) The HTTP based messaging is always initiated by the client:

- HTTP request is generated by the client (i.e. UE);
- HTTP response is generated by the server as a response to the HTTP request;
- HTTP proxies may be between the client and the server.

b) There is only one single HTTP response to the HTTP request.

c) In order to differentiate between HTTP messages and other protocol messages, the HTTP messages are marked with simple arrow line, and all *non-HTTP* messages with block arrows.

d) The flows show the signalling exchanges between the following functional entities in addition to those specified in 3GPP TS 24.228 [13]:

- Bootstrapping Server Function (BSF);
- Network Application Function (NAF);
- PKI portal (PKI portal).

e) The "(B-TID)" sequence of characters is used to indicate that the bootstrapping transaction identifier (B-TID) needs to be filled in.

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.

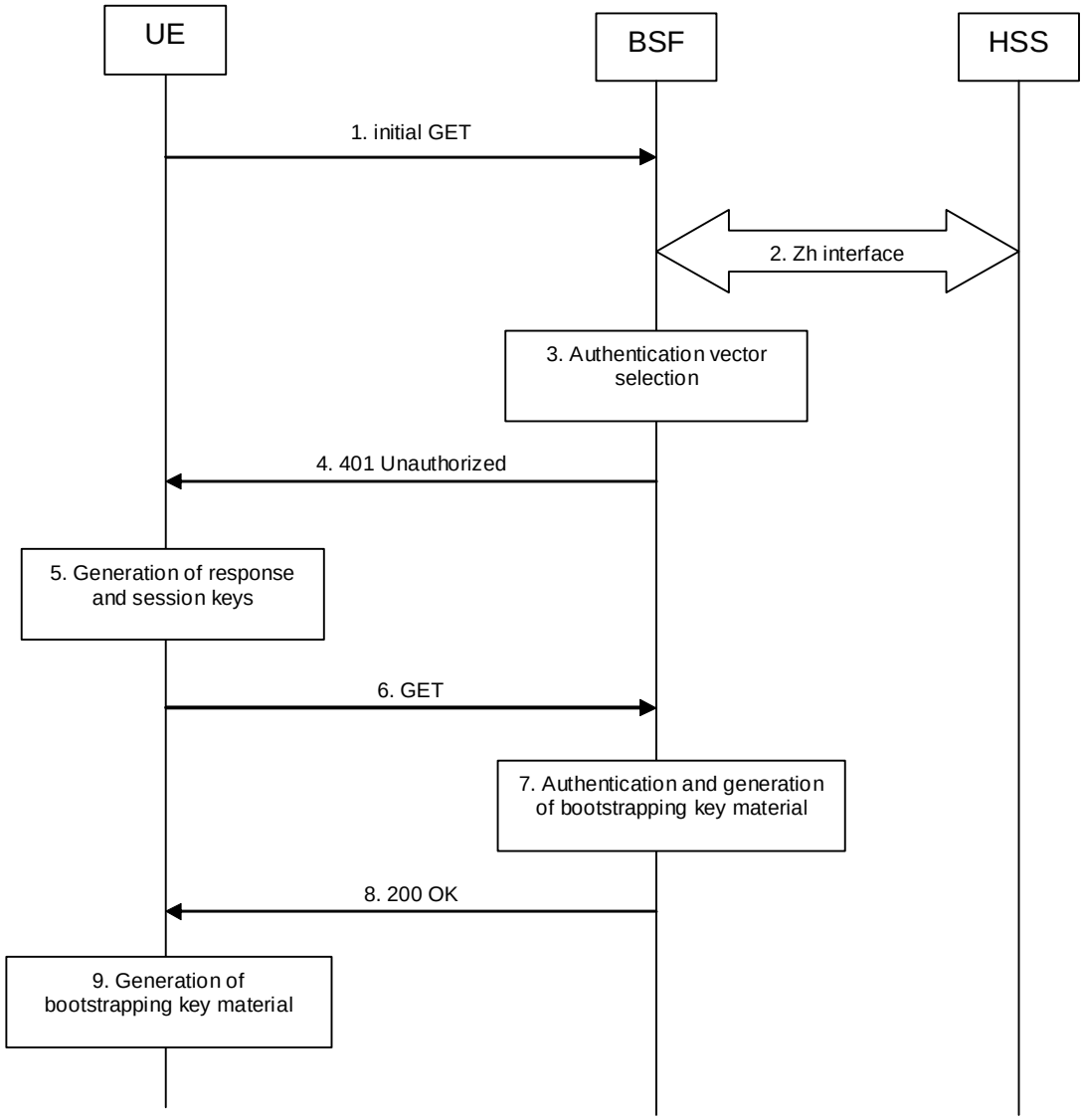

Figure A.3-1: Bootstrapping signalling

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. If the UE does not have an IMS subscription, the private user identity will be constructed from IMSI, according to 3GPP TS 23.003 [6].

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 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 [20].

NOTE 1: The authentication vector may be of the form as in 3GPP TS 33.203 [20] (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.

Table A.3-3: Void

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

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 [5].

Table A.3-4: 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 may 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 response, RES, and also computes the session keys IK and CK. The RES is put into the Authorization header and sent back to the registrar in the GET request.

Table A.3-5: Void

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

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

Table A.3-6: 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 11 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 response, RES, the BSF checks that the user's active, XRES matches the received RES. 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].

Table A.3-7: Void

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

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

Table A.3-8: 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"?>
<bsf xmlns="urn-to-xml-schema-of-3gpp-bsf"
    lifetime="NNNN"
/>
```

Editor's note: The detailed B-TID transport mechanism is FFS

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 shall include the rspauth parameter which is calculated as specified in RFC 2617 [8] using RES for response calculation as specified in RFC 3310 [5].

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].

Table A.3-9: Void

A.4 Signalling flows demonstrating a synchronization failure in the bootstrapping procedure

If the UE considers the sequence number in the challenge to be not in the correct range, it shall send a synchronization failure indication back to BSF. The parameter AUTS contains the concealed value of the counter value SQN_{MS} in the UE.

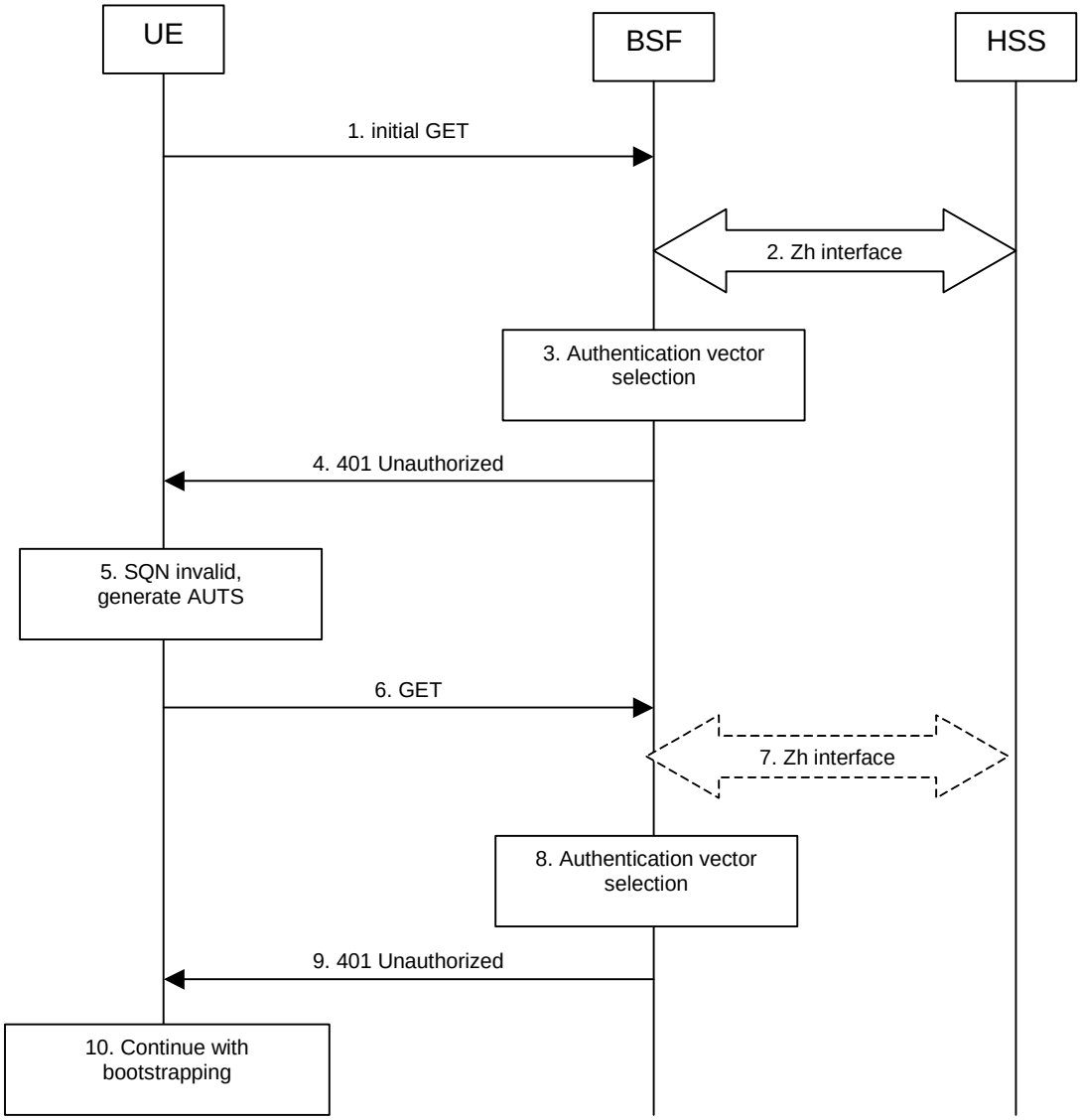

Figure A.4-1: The bootstrapping procedure in sequence number synchronization failure case.

1-4. **Initial bootstrapping steps**

Steps 1 through 4 are described in the corresponding steps in clause A.3.

Tables A.4-1 to A.4-4: Void

5. **SQN invalid, generate AUTS at UE**

The UE identifies the sequence number is out of synchronization. The UE shall generate the AUTS parameter (112 bit value). The AUTS parameter is populated in Authorization header, as specified in RFC 3310 [5].

Table A.4-5: Void

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

The UE sends HTTP GET request, with the AUTS parameter to the BSF.

Table A.4-6: 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=base64(RAND + AUTN + server specific data), uri="/", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=AKAv1-MD5, auts=base64(AUTS)
```

Authorization: This carries the response to the authentication challenge received in step 4 and contains the AUTS parameter.

7. **Zh: Authentication procedure**

   If BSF does not have the corresponding AV indicated by the AUTS, the BSF shall retrieve it from the HSS.

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

Table A.4-7: 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.

8. **Authentication vector selection**

   The BSF selects the AV indicated by the AUTS for use in the authentication challenge. For detailed description of the authentication vector, see 3GPP TS 33.203 [20].

Table A.4-8: Void

9. **401 Unauthorized response (BSF to UE) - see example in table A.4-9**

   The BSF shall send another challenge based on new range of sequence number.

Table A.4-9: 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"
```

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

10. **Continue with bootstrapping**

    The bootstrapping procedure continues from step 5 of clause A.3.

Table A.4-10: Void

Annex B (informative): Signalling flows for HTTP Digest Authentication with bootstrapped security association

B.1 Scope of signalling flows

This annex gives examples of signalling flows for using HTTP Digest Authentication with bootstrapped security association.

B.2 Introduction

B.2.1 General

A bootstrapping session established using a bootstrapping procedure (cf. clause 4 and annex A) is used between a UE and a NAF. The BSF provides to the NAF a NAF specific key material (Ks_NAF or Ks_ext_NAF) which is derived from the key material (Ks or Ks_ext). The NAF uses this key to authenticate and optionally secure (i.e. integrity protect and encrypt) the communications between it and the UE. The BSF will also provide the NAF the expiration time of the bootstrapping session. When the bootstrapping session becomes invalid the NAF will stop using the session, and indicate to the UE that bootstrapping session has expired and that new session needs to be established.

An example of the signalling flows of the authentication procedure using HTTP Digest authentication [8] is given in clause B.3.

B.2.2 Key required to interpret signalling flows

The key to interpret signalling flows is specified in subclause A.2.2.

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 may take place inside a server-authenticated TLS (as described in RFC 2246 [10]) tunnel in which case TLS session has been established before step 1.

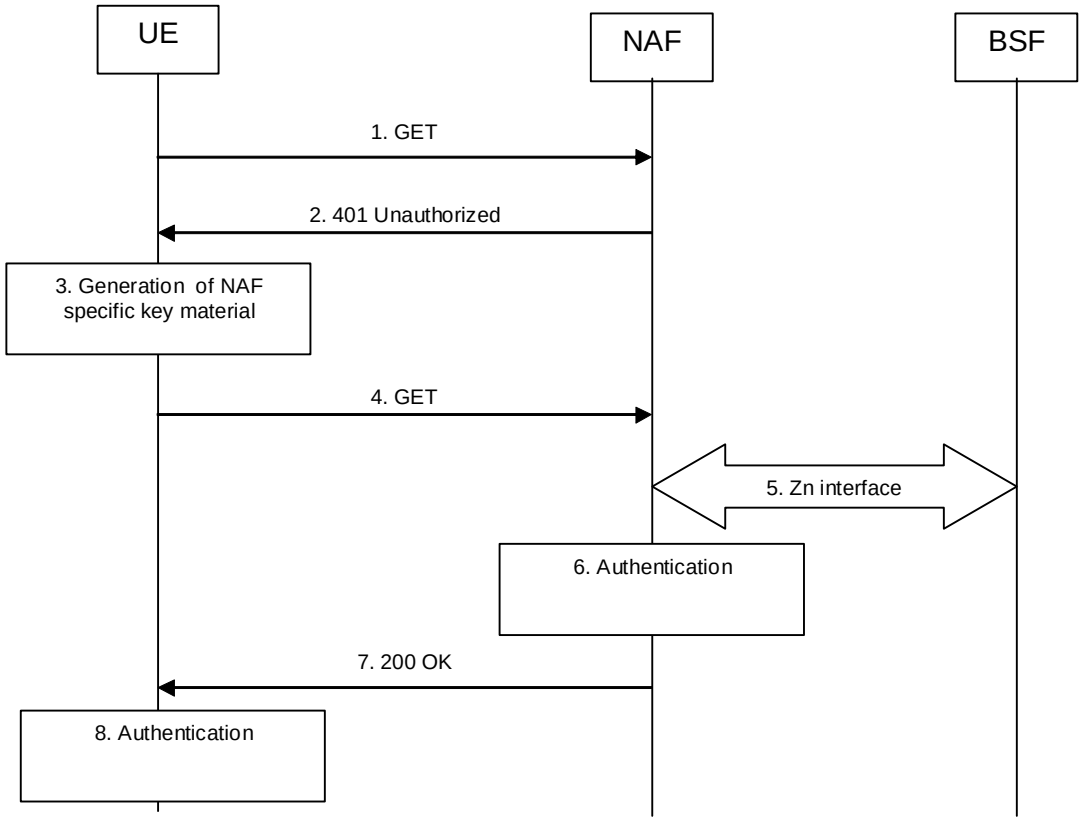

Figure B.3-1: HTTP Digest Authentication with bootstrapped security association

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 shall include 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 may also be a POST request in which case the request would contain a client payload in the HTTP response and the corresponding Content-Type and Content-Length header values.

2. **401 Unauthorized response (NAF to UE) - see example in table B.3.1-2**

Upon receiving an HTTP request that contains static string "3gpp-gba" in the User-Agent header, NAF may 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 may 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 shall verify 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 will obtain one by running bootstrapping procedure over Ub interface

Table B.3-3: Void

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

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.3-4: 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 shall be set to "auth-int" by default. If the conversation is taking place inside a server-authenticated TLS tunnel, the qop attribute may 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 response 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.3.1-5: 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 shall also verify that the DNS name in the realm attribute matches its own. If the conversation is taking place inside a server-authenticated TLS tunnel, the NAF shall also verify that this DNS name is the same as that of the TLS server.

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

Table B.3-6: Void

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

The BSF 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 may use key material Ks_NAF to integrity protect and authenticate the response.

Table B.3-7: 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 and responses must be constructed according to RFC 2617 [8].

Table B.3-8: Void

Annex C (normative): XML Schema Definition

C.1 Introduction

This annex contains the XML schema definition for an XML document carrying the bootstrapping key lifetime, and possibly other server specific data.

Editor's note: It is FFS whether the B-TID will be carried in the XML document.

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="urn-to-xml-schema-of-3gpp-bsf"
    xmlns:tns="urn-to-xml-schema-of-3gpp-bsf"
    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:element name="bsf" type="tns:bsf"/>

  <xs:complexType name="bsf">
    <xs:sequence>
      <xs:attribute name="lifetime" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

Annex D (informative): Signalling flows for Authentication Proxy

D.1 Scope of signalling flows

This annex gives examples of signalling flows for using AP in GAA.

D.2 Introduction

D.2.1 General

Void

D.2.2 Key required to interpret signalling flows

The key to interpret signalling flows specified in subclause A.2.2.

D.3 Signalling flow demonstrating a successful authentication procedure

Editor's note: An example of authentication procedure with AP needs be added.

Annex E (informative): Signalling flows for PKI portal

E.1 Scope of signalling flows

This annex gives examples of signalling flows for the subscriber certificate enrolment and the CA certificate delivery.

E.2 Introduction

E.2.1 General

A bootstrapping session established using a bootstrapping procedure (cf., clause 4 and annex A) is used between a UE and a PKI portal. The BSF provides to the PKI portal a NAF specific key material (Ks_NAF or Ks_ext_NAF) which is derived from the key material (Ks or Ks_ext). The PKI portal uses this key to authenticate and optionally secure (i.e. integrity protect and encrypt) the communications between it and the UE. The BSF will also provide the PKI portal the expiration time of the bootstrapping session.

E.2.2 Key required to interpret signalling flows

The key to interpret signalling flows is specified in subclause A.2.2.

E.3 Signalling flows demonstrating a successful subscriber certificate enrolment

E.3.1 Simple subscriber certificate enrolment

The signalling flow in figure E.3.1-1 describes the message exchange between UE and PKI portal when UE wants to enrol a subscriber certificate. The messaging may take place inside a server-authenticated TLS (as described in RFC 2246 [10]) tunnel in which case TLS session has been established before step 1.

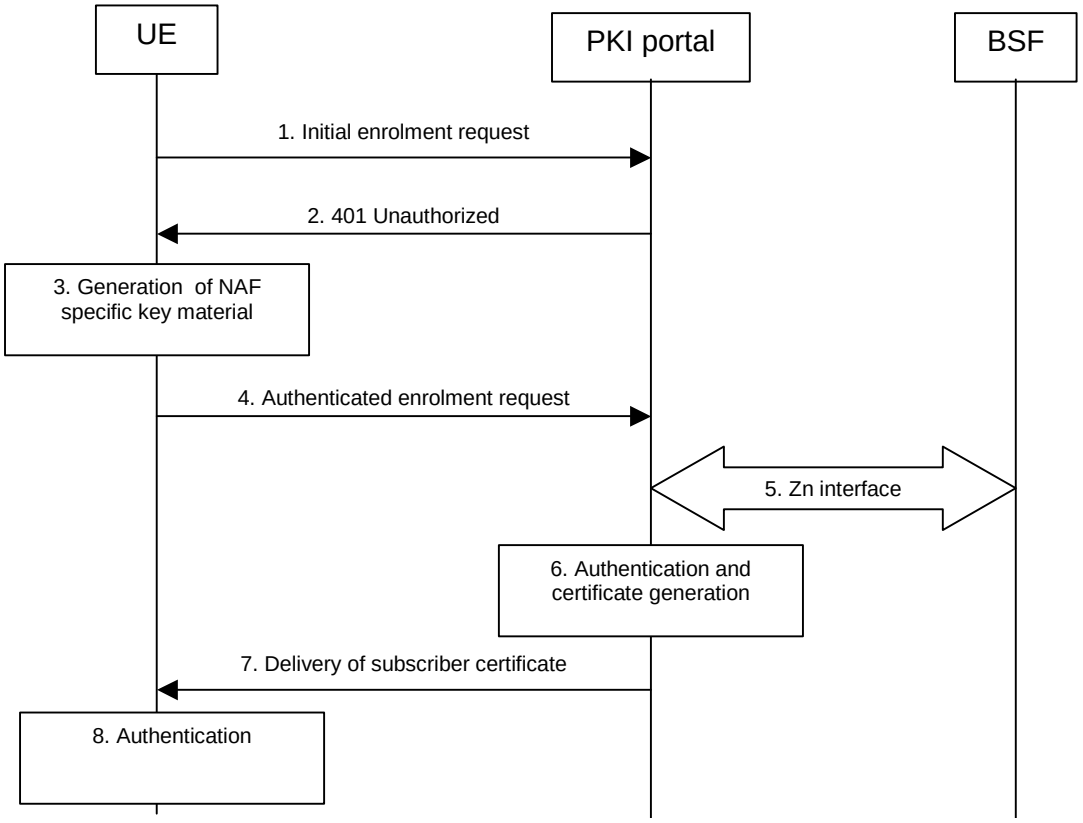

Figure E.3.1-1: Successful subscriber certificate enrolment.

1. **Initial enrolment request (UE to PKI portal) - see example in table E.3.1-1**

The UE sends an HTTP request to the PKI portal containing a PKCS#10 certification request.

Table E.3.1-1: Initial enrolment request (UE to PKI portal)

```
POST /enrol?response=single HTTP/1.1
Host: pkiportal.home1.net:1234
Content-Type: application/x-pkcs10
Content-Length: (...)
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referrer: http://pkiportal.home1.net:1234/service

----- BEGIN CERTIFICATE REQUEST -----
<PKCS#10 BLOB>
----- END CERTIFICATE REQUEST -----
```

Request-URI: The Request-URI (the URI that follows the method name, "POST", in the first line) indicates the resource of this POST request. The Request-URI contains the parameter "response" which is set to "single" to indicate to the PKI portal the desired response type, i.e. just the subscriber certificate is requested to be delivered.

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

Content-Type: Contains the media type "application/x-pkcs10", i.e. the PKCS#10.

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

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 URI for the PKI portal was obtained.

NOTE 1: This step is used to trigger the GBA-based authentication between the UE and the PKI portal.

2. **401 Unauthorized response (PKI portal to UE) - see example in table E.3.1-2**

The PKI portal 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 E.3.1-2: 401 Unauthorized response (PKI portal 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@pkiportal.home1.net",
nonce="6629fae49393a05397450978507c4ef1", algorithm=MD5, qop="auth,auth-int",
opaque="5ccc069c403ebaf9f0171e9517f30e41"
```

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

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

WWW-Authenticate: The PKI portal 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 messaging is taking place inside a server-authenticated TLS tunnel, the options for the qop attribute may 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 hostname of the server (i.e. FQDN of the PKI portal).

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

The UE shall verify that the second part of the realm attribute does correspond to the server it is talking to. In particular, if the messaging is taking place inside a server-authenticated TLS tunnel, the UE shall verify that the server name (i.e. FQDN of the PKI portal) in the server's TLS certificate matches the hostname of the server 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 will obtain one by running bootstrapping procedure over Ub interface.

Table E.1-3: Void

4. **Authenticated enrolment request (UE to PKI portal) - see example in table E.3.1-4**

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 PKI portal.

Table E.3.1-4: Authenticated enrolment request (UE to PKI portal)

```
POST /enrol?response=single HTTP/1.1
Host: pkiportal.home1.net:1234
Content-Type: application/pkcs10
Content-Length: (...)
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://pkiportal.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@pkiportal.home1.net",
nonce="a6332ffd2d234==", uri="/enrol?response=single", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=MD5

----- BEGIN CERTIFICATE REQUEST -----
<PKCS#10 BLOB>
----- END CERTIFICATE REQUEST -----
```

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 shall be set to "auth-int" by default. If the messaging is taking place inside a server-authenticated TLS tunnel, the qop attribute may 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**

PKI portal retrieves the NAF specific key material (Ks_NAF) and subscriber's user security setting from the BSF.

NOTE 4: Subscriber's user security setting for PKI portal consists of flags that indicate whether certain type certificate is authorized to be issued to the subscriber. There are two certificate types: authentication certificate and non-repudiation certificate.

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

Table E.3.1-5: Bootstrapping authentication information procedure (PKI portal 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 and certificate generation at PKI portal**

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

The PKI portal shall also verify that the hostname (i.e. its FQDN) in the realm attribute matches its own. If the messaging is taking place inside a server-authenticated TLS tunnel, the PKI portal shall also verify that this hostname is the same as that of the TLS server.

If the verification succeeds, the incoming client-payload request is taken in for further processing. The PKI portal continues processing of the PKCS#10 request according to its internal policies. The PKI portal shall verify that the subscriber is allowed to receive the particular type of certificate indicate in the PKCS#10 request by checking subscriber's user security setting received from the BSF in step 5.

NOTE 5: The procedures for generating the subscriber certificate are outside the scope.

Table B.3.1-6: Void

7. **Delivery of subscriber certificate (PKI portal to UE) - see example in table E.3.1-7**

The PKI portal sends 200 OK response to the UE to indicate the success of the authentication and the subscriber certificate enrolment. The PKI portal generates a HTTP response containing the enrolled subscriber certificate. The PKI portal may use key material Ks_NAF to integrity protect and authenticate the response.

Table E.3.1-7: Delivery of subscriber certificate (PKI portal to UE)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/html
Content-Type: application/x-x509-user-cert
Content-Length: (...)
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

----- BEGIN CERTIFICATE -----
<Subscriber certificate BLOB>
----- END CERTIFICATE -----
```

Content-Type: Contains the media type "application/x-x509-user-cert", i.e. X.509 certificate.

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**

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

Table E.3.1-8: Void

E.3.2 Subscriber certificate enrolment with WIM authentication codes

The signalling flow in figure E.3.2-1 describes the message exchange between UE and PKI portal when UE wants to enrol a subscriber certificate, and the UE uses a WIM that requires authentication codes both for onboard key pair generation and proof-of-origin generation. The messaging may take place inside a server-authenticated TLS (as described in RFC 2246 [10]) tunnel in which case TLS session has been established before step 1.

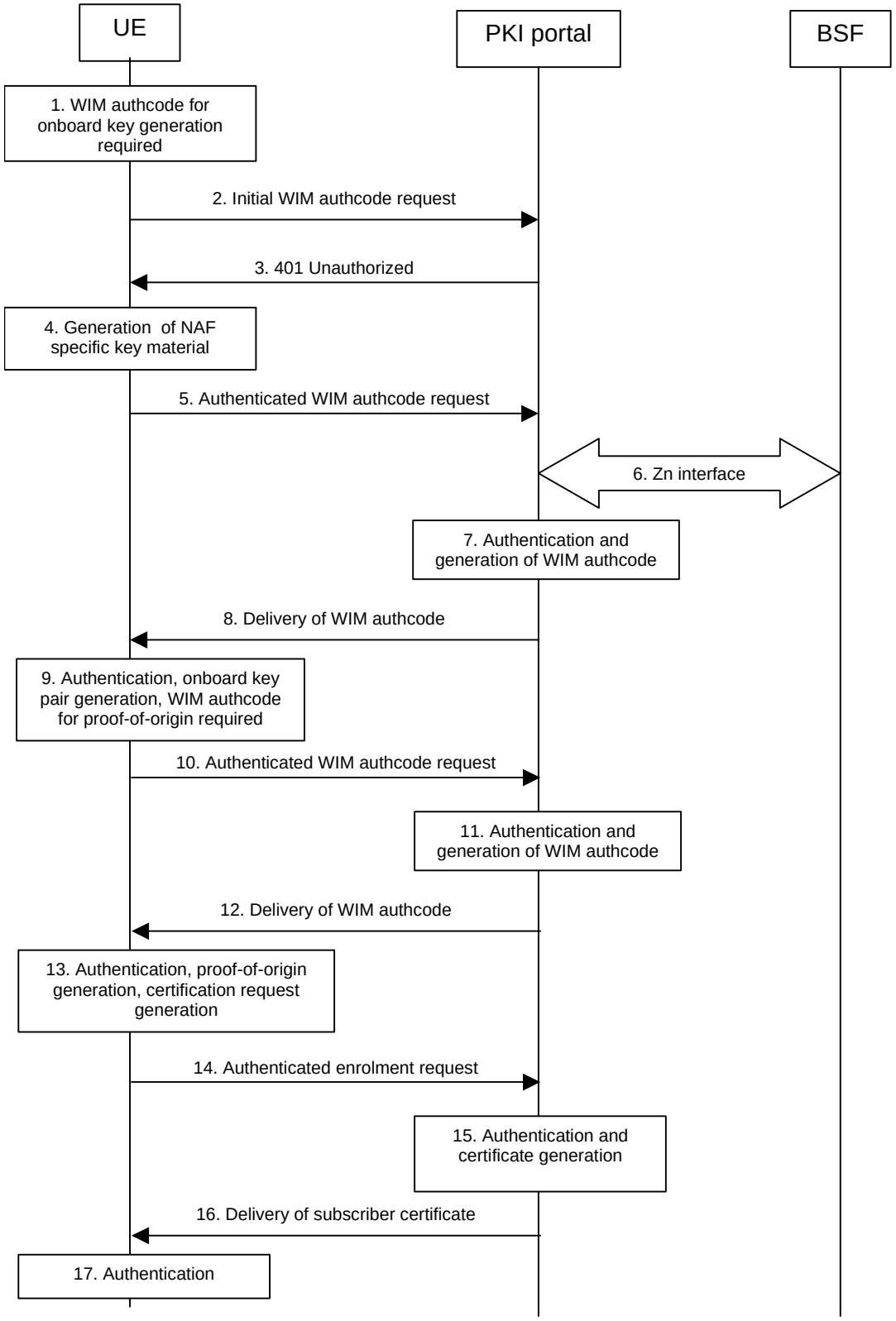

Figure E.3.2-1: Successful subscriber certificate enrolment

1. **WIM authentication code for onboard key pair generation required at UE**

   The UE has initiated enrolment procedure and the WIM in the UE requires an WIM authentication code for the onboard key pair generation.

NOTE 1: It is not mandatory to generate a key pair for each enrolment procedure, and the WIM may not require WIM authentication code for generating the key pair. In these cases, the WIM authentication code is not needed.

Table E.3.2-1: Void

2. **Initial WIM authentication code request (UE to PKI portal) - see example in table E.3.2-2**

   The UE sends an HTTP request to the PKI portal containing a WIM authentication code request.

Table E.3.2-2: Initial WIM authentication code request (UE to PKI portal)

```
GET /enrol/wim-auth-code?request=error:AuthReq:123456789ABCDEF:AABBCCDDEE HTTP/1.1
Host: pkiportal.home1.net:1234
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referrer: http://pkiportal.home1.net:1234/service
```

Request-URI: The Request-URI (the URI that follows the method name, "GET", in the first line) indicates the resource of this GET request. The Request-URI contains the parameter "request" which contains the WIM authentication request parameter received from the WIM, i.e. a static string "error:AuthReq:" appended by the WIM serial number in hexadecimal format, colon ":", and the challenge data in hexadecimal format.

Host: Specifies the Internet host and port number of the PKI portal 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 URI for the PKI portal was obtained.

NOTE 2: This step is used to trigger the GBA-based authentication between the UE and the PKI portal.

3. **401 Unauthorized response (PKI portal to UE) - see example in table E.3.2-3**

   The PKI portal 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 E.3.2-3: 401 Unauthorized response (PKI portal 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@pkiportal.home1.net",
nonce="6629fae49393a05397450978507c4ef1", algorithm=MD5, qop="auth,auth-int",
opaque="5ccc069c403ebaf9f0171e9517f30e41"
```

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

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

WWW-Authenticate: The PKI portal 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 messaging is taking place inside a server-authenticated TLS tunnel, the options for the qop attribute may 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 hostname of the server (i.e. FQDN of the PKI portal).

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

The UE shall verify that the second part of the realm attribute does correspond to the server it is talking to. In particular, if the messaging is taking place inside a server-authenticated TLS tunnel, the UE shall verify that the server name (i.e. FQDN of the PKI portal) in the server's TLS certificate matches the hostname of the server 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 3: If UE does not have a bootstrapped security association available, it will obtain one by running bootstrapping procedure over Ub interface.

Table E.3.2-4: Void

5. **Authenticated WIM authentication code request (UE to PKI portal) - see example in table E.3.2-5**

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 as the password, and sends the request to PKI portal.

Table E.3.2-5: Authenticated WIM authentication code request (UE to PKI portal)

```
GET /enrol/wim-auth-code?request=error:AuthReq:123456789ABCDEF:AABBCCDDEE HTTP/1.1
Host: pkiportal.home1.net:1234
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://pkiportal.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@pkiportal.home1.net",
nonce="a6332ffd2d234==", uri="/enrol/wim-auth-code?request=error:AuthReq:123456789ABCDEF:AABBCCDDEE
", 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 shall be set to "auth-int" by default. If the messaging is taking place inside a server-authenticated TLS tunnel, the qop attribute may be set to "auth" as well.

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

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

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

Table E.3.2-6: Bootstrapping authentication information procedure (PKI portal 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.

7. **Authentication and WIM authentication code generation at NAF**

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

The PKI portal shall also verify that the hostname (i.e. its FQDN) in the realm attribute matches its own. If the messaging is taking place inside a server-authenticated TLS tunnel, the PKI portal shall also verify that this hostname is the same as that of the TLS server.

If the verification succeeds, the WIM authentication code is taken in for further processing. The PKI portal continues processing of the WIM authentication code request according to its internal policies.

NOTE 4: The procedures for generating the WIM authentication code are outside the scope.

Table E.3.2-7: Void

8. **Delivery of WIM authentication code (PKI portal to UE) - see example in table E.3.2-8**

The PKI portal sends 200 OK response to the UE to indicate the success of the authentication and the WIM authentication code generation. The PKI portal generates a HTTP response containing the WIM authentication code. The PKI portal may use key material Ks_NAF to integrity protect and authenticate the response.

Table E.3.2-8: Delivery of WIM authentication code (PKI portal to UE)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/plain
Content-Length: (...)
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

13579BDF2468ACE
```

Content-Type: Contains the media type "text/plain".

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.

9. **Authentication, key pair generation, and WIM authentication code request for proof-of-origin generation at UE**

The UE receives the response and verifies the Authentication-Info header. If the verification succeeds, the UE can use the WIM authentication code in the onboard key pair generation with the WIM.

The WIM in the UE also requires a WIM authentication code for the proof-of-origin generation.

NOTE 5: It is not mandatory to include the proof-of-origin to certificate request of the enrolment procedure, and the WIM may not require WIM authentication code for generating the proof-of-origin. In these cases, the WIM authentication code is not needed.

Table E.3.2-9: Void

10. **Authenticated WIM authentication code request (UE to PKI portal) - see example in table E.3.2-10**

The 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 as the password, and sends the request to PKI portal.

Table E.3.2-10: Authenticated WIM authentication code request (UE to PKI portal)

```
GET /enrol/wim-auth-code?request=error:AuthReq:1122334455667788:1122334455 HTTP/1.1
Host: pkiportal.home1.net:1234
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://pkiportal.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@pkiportal.home1.net",
nonce="a6332ffd2d234==", uri="/enrol/wim-auth-code?request=error:AuthReq:123456789ABCDEF:AABBCCDDEE
", qop=auth-int, nc=00000001, cnonce="6629fae49393a05397450978507c4ef1",
response="6629fae49393a05397450978507c4ef1, opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=MD5
```

11. **Authentication and WIM authentication code generation at NAF**

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

The PKI portal shall also verify that the hostname (i.e. its FQDN) in the realm attribute matches its own. If the messaging is taking place inside a server-authenticated TLS tunnel, the PKI portal shall also verify that this hostname is the same as that of the TLS server.

If the verification succeeds, the WIM authentication code is taken in for further processing. The PKI portal continues processing of the WIM authentication code request according to its internal policies.

NOTE 6: The procedures for generating the WIM authentication code are outside the scope.

Table E.3.2-11: Void

12. **Delivery of WIM authentication code (PKI portal to UE) - see example in table E.3.2-12**

The PKI portal sends 200 OK response to the UE to indicate the success of the authentication and the WIM authentication code generation. The PKI portal generates a HTTP response containing the WIM authentication code. The PKI portal may use key material Ks_NAF to integrity protect and authenticate the response.

Table E.3.2-12: Delivery of WIM authentication code (PKI portal to UE)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/plain
Content-Length: (...)
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

FFEEDDCCBBAA998877665544
```

Content-Type: Contains the media type "text/plain".

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.

13. **Authentication, proof-key-origin key pair generation, and PKCS#10 generation at UE**

The UE receives the response and verifies the Authentication-Info header. If the verification succeeds, the UE can use the WIM authentication code in the proof-of-origin generation with the WIM.

The WIM in the UE also requires a WIM authentication code for the proof-of-origin generation.

NOTE 7: It is not mandatory to include the proof-of-origin to certificate request of the enrolment procedure, and the WIM may not require WIM authentication code for generating the proof-of-origin. In these cases, the WIM authentication code is not needed.

Table E.3.2-13: Void

14. **Authenticated enrolment request (UE to PKI portal) - see example in table E.3.2-14**

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 as the password, and sends the request to PKI portal.

Table E.3.2-14: Authenticated enrolment request (UE to PKI portal)

```
POST /enrol?response=single HTTP/1.1
Host: pkiportal.home1.net:1234
Content-Type: application/pkcs10
Content-Length: (...)
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://pkiportal.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@pkiportal.home1.net",
nonce="a6332ffd2d234==", uri="/enrol?response=single", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1,
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=MD5

----- BEGIN CERTIFICATE REQUEST -----
<PKCS#10 BLOB>
----- END CERTIFICATE REQUEST -----
```

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

15. **Authentication and certificate generation at PKI portal**

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

The PKI portal shall also verify that the hostname (i.e. its FQDN) in the realm attribute matches its own. If the messaging is taking place inside a server-authenticated TLS tunnel, the PKI portal shall also verify that this hostname is the same as that of the TLS server.

If the verification succeeds, the incoming client-payload request is taken in for further processing. The PKI portal continues processing of the PKCS#10 request according to its internal policies.

NOTE 8: The procedures for generating the subscriber certificate are outside the scope.

Table E.3.2-15: Void

16. **Delivery of subscriber certificate (PKI portal to UE) - see example in table E.3.2-16**

The PKI portal sends 200 OK response to the UE to indicate the success of the authentication and the subscriber certificate enrolment. The PKI portal generates a HTTP response containing the enrolled subscriber certificate. The PKI portal may use key material Ks_NAF to integrity protect and authenticate the response.

Table E.3.2-16: Delivery of subscriber certificate (PKI portal to UE)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/html
Content-Type: application/x-x509-user-cert
Content-Length: (...)
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

----- BEGIN CERTIFICATE -----
<Subscriber certificate BLOB>
----- END CERTIFICATE -----
```

Content-Type: Contains the media type "application/x-x509-user-cert", i.e. X.509 certificate.

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.

17. **Authentication at UE**

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

Table E.3.2-17: Void

E.4 Signalling flows demonstrating a failure in subscriber certificate enrolment

The signalling flow in figure E.3.1-1 describes the message exchange between UE and PKI portal using HTTP Digest Authentication. This clause describes a failure in the subscriber certificate enrolment, related to PKI procedures. Thus, it assumed that subscriber certificate enrolment procedure has proceeded to step 6 as described in subclause E.3.1.

Tables E.4-1 to E.4-5: Void

6. **Authentication and certificate generation at PKI portal**

The verification procedures described in subclause E.3.1 step 6 are successfully completed.

The PKI portal encounters an error during the internal enrolment procedure. For example, the PKI portal is not allowed to issue a certificate to the subscriber due operator's internal policies, i.e. the subscriber's profile in the HSS indicates that the enrolment is not allowed.

Table E.4-6: Void

7. **Error notification (PKI portal to UE) - see example in table E.4-7**

   The PKI portal sends 403 Forbidden response to the UE to indicate that the subscriber certificate enrolment is allowed. The PKI portal generates a HTTP response containing the error notification. The PKI portal may use key material Ks_NAF to authenticate the response.

Table E.4-7: Error notification (PKI portal to UE)

```
HTTP/1.1 403 Forbidden
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/html
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
```

Authentication-Info: This carries the protection

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

8. **Authentication at UE**

   The UE receives the response and verifies the Authentication-Info header. If the verification succeeds, the UE is notified of the failure of the subscriber certificate enrolment.

Table E.4-8: Void

E.5 Signalling flows demonstrating a successful CA certificate delivery

The signalling flow in figure E.5-1 describes the message exchange between UE and PKI portal when UE requests a CA certificate delivery. The messaging may take place inside a server-authenticated TLS (as described in RFC 2246 [10]) tunnel in which case TLS session has been established before step 1.

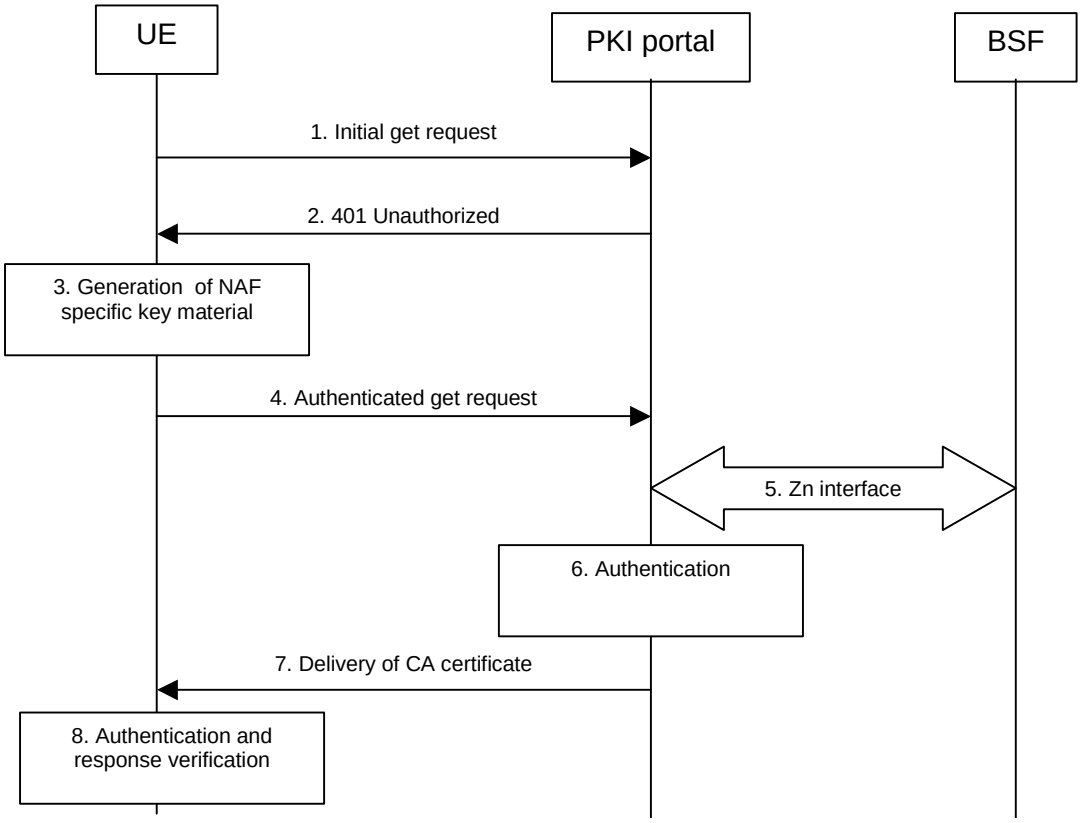

Figure E.5-1: Successful CA certificate delivery.

1. **Initial get request (UE to PKI portal) - see example in table E.5-1**

The UE sends an HTTP request to the PKI portal requesting the delivery of CA certificate.

Table E.5-1: Initial get request (UE to PKI portal)

```
GET /getcertificate?in=aabbccdd== HTTP/1.1
Host: pkiportal.home1.net:1234
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referrer: http://pkiportal.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. The Request-URI contains the parameter "in" (i.e. issuer name) which is set to the Base64 encoding of the DER encoded Issuer field of the X.509 certificate.

Host: Specifies the Internet host and port number of the PKI portal 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 URI for the PKI portal was obtained.

NOTE 1: This step is used to trigger the GBA-based authentication between the UE and the PKI portal.

2. **401 Unauthorized response (PKI portal to UE) - see example in table E.5.1-2**

The PKI portal 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 E.5-2: 401 Unauthorized response (PKI portal 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@pkiportal.home1.net",
nonce="6629fae49393a05397450978507c4ef1", algorithm=MD5, qop="auth,auth-int",
opaque="5ccc069c403ebaf9f0171e9517f30e41"
```

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

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

WWW-Authenticate: The PKI portal 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 messaging is taking place inside a server-authenticated TLS tunnel, the options for the qop attribute may 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 host of the server (i.e. the FQDN of the PKI portal).

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

The UE shall verify that the second part of the realm attribute does correspond to the server it is talking to. In particular, if the messaging is taking place inside a server-authenticated TLS tunnel, the UE shall verify that the server name (i.e. FQDN of the PKI portal) in the server's TLS certificate matches the hostname of the server 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 will obtain one by running bootstrapping procedure over Ub interface.

Table E.5-3: Void

4. **Authenticated get request (UE to PKI portal) - see example in table E.5-4**

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 PKI portal.

Table E.5-4: Authenticated get request (UE to PKI portal)

```
GET /getcertificate?in=aabbccdd== HTTP/1.1
Host: pkiportal.home1.net:1234
User-Agent: SCEnrolmentAgent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://pkiportal.home1.net:1234/service
Authorization: Digest username="(B-TID)", realm="3GPP-bootstrapping@pkiportal.home1.net",
nonce="a6332ffd2d234==", uri="/getcertificate?in=aabbccdd==", 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 shall be set to "auth-int" by default. If the messaging is taking place inside a server-authenticated TLS tunnel, the qop attribute may be set to "auth" as well.

NOTE 3: If step 1 was a GET request then this request would also be GET request and contain the same Request-URI in the HTTP request as was carried in step 1.

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

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

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

Table E.5.1-5: Bootstrapping authentication information procedure (PKI portal 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 PKI portal**

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

The PKI portal shall also verify that the hostname (i.e. its FQDN) in the realm attribute matches its own. If the HTTP messaging is taking place inside a server-authenticated TLS tunnel, the PKI portal shall also verify that this hostname is the same as that of the TLS server.

If the verification succeeds, the incoming client-payload request is taken in for further processing, i.e. the PKI portal shall send the requested CA certificate to the UE.

Table E.5-6: Void

7. **Delivery of CA certificate (PKI portal to UE) – see example in table E.5-7**

The PKI portal sends 200 OK response to the UE to indicate the success of the authentication. The PKI portal generates a HTTP response containing the requested CA certificate. The PKI portal shall use the key material Ks_NAF to integrity protect and authenticate the response.

Table E.5-7: Delivery of CA certificate (PKI portal to UE)

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/html
Content-Type: application/x-x509-ca-cert
Content-Length: (...)
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

----- BEGIN CERTIFICATE -----
<CA certificate BLOB>
----- END CERTIFICATE -----
```

Content-Type: Contains the media type "application/x-x509-ca-cert", i.e. X.509 CA certificate.

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 and response verification at UE**

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

Table E.5-8: Void

E.6 Signalling flows demonstrating a failure in CA certificate delivery

The signalling flow in figure E.5-1 describes the message exchange between UE and PKI portal when UE requests a CA certificate delivery. This clause describes a failure in the CA certificate delivery. It assumed that CA certificate delivery procedure has proceeded to step 6 as described in clause E.5.

Tables E.6-1 to E.6-5: Void

6. **Authentication at PKI portal**

   The verification procedures described in clause E.5 step 6 are successfully completed.

   The PKI portal discovers that it does not have the requested CA certificate.

Table E.6-6: Void

7. **Error notification (PKI portal to UE) - see example in table E.6-7**

   The PKI portal sends 404 Not Found response to the UE to indicate that the requested CA certificate is not found in the PKI portal. The PKI portal may use key material Ks_NAF to authenticate the response.

Table E.6-7: Error notification (PKI portal to UE)

```
HTTP/1.1 404 Not Found
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Content-Type: text/html
Authentication-Info: qop=auth-int, rspauth="6629fae49394a05397450978507c4ef1",
cnonce="6629fae49393a05397450978507c4ef1", nc=00000001
Date: Thu, 08 Jan 2004 10:50:35 GMT
```

Authentication-Info: This carries the protection

8. **Authentication and response verification at UE**

   The UE receives the response and verifies the Authentication-Info header. If the verification succeeds, the UE is notified of the failure of the CA certificate delivery.

Table E.6-8: Void

Annex F (informative): Signalling flows for PSK TLS with bootstrapped security association

Editor's note: If the "Pre-Shared Key Ciphersuites for TLS" Internet Draft [14A] does not reach the RFC status by the time when Release 6 is frozen, this annex shall be removed and the support for the Pre-Shared Key TLS is postponed to Release 7.

F.1 Scope of signalling flows

This annex gives examples of signalling flows for using PSK TLS with bootstrapped security association.

F.2 Introduction

F.2.1 General

A bootstrapping session established using a bootstrapping procedure (cf., clause 4 and annex A) is used between a UE and a NAF. The BSF provides to the NAF a NAF specific key material (Ks_NAF or Ks_ext_NAF) which is derived from the key material (Ks or Ks_ext). The NAF uses this key to authenticate and optionally secure (i.e. integrity protect and encrypt) the communications between it and the UE. The BSF will also provide the NAF the expiration time of the bootstrapping session. When the bootstrapping session becomes invalid the NAF will stop using the session, and indicate to the UE that bootstrapping session has expired and that new session needs to be established.

An example of the signalling flows of the authentication procedure using PSK TLS [14A] is given in clause F.3.

F.2.2 Key required to interpret signalling flows

The following key (rules) have been applied to TLS handshake signalling flows to improve readability, reduce errors and increase maintainability:

a) The description of TLS messages and their fields are identified by three fields: "TLS.MESSAGE.FIELD":

- "TLS" identifies that the message is a TLS message;
- "MESSAGE" identifies the name of the TLS message (e.g. ClientHello);
- "FIELD" identifies the name of the TLS message field (e.g. client_version).

An example being "TLS.ClientHello.client_version", which identifies TLS message "ClientHello" and its data field "client_version". The possible TLS message and TLS message field names as well as their encoding to the TLS protocol are specified in IETF TLS related specifications such as IETF RFC 2246 [10] and IETF RFC 3546 [17].

b) If multiple TLS messages are sent in sequence from one entity to another this is described as one step.

- the figures describe the sending of multiple TLS messages in one step by listing the TLS message names in separate lines;
- the description of the step contains the explanation of the messages and their parameters as described in bullet a).

c) In order to differentiate between TLS messages and other protocol messages, the TLS messages are marked with simple arrow line, and all *non-TLS* messages with block arrows.

d) The flows show the signalling exchanges between the following functional entities:

- User Equipment (UE);
- Bootstrapping Server Function (BSF);
- Network Application Function (NAF).

F.3 Signalling flow demonstrating a successful authentication procedure

The signalling flow in figure F.3-1 describes the generic message exchange between UE and NAF using PSK TLS.

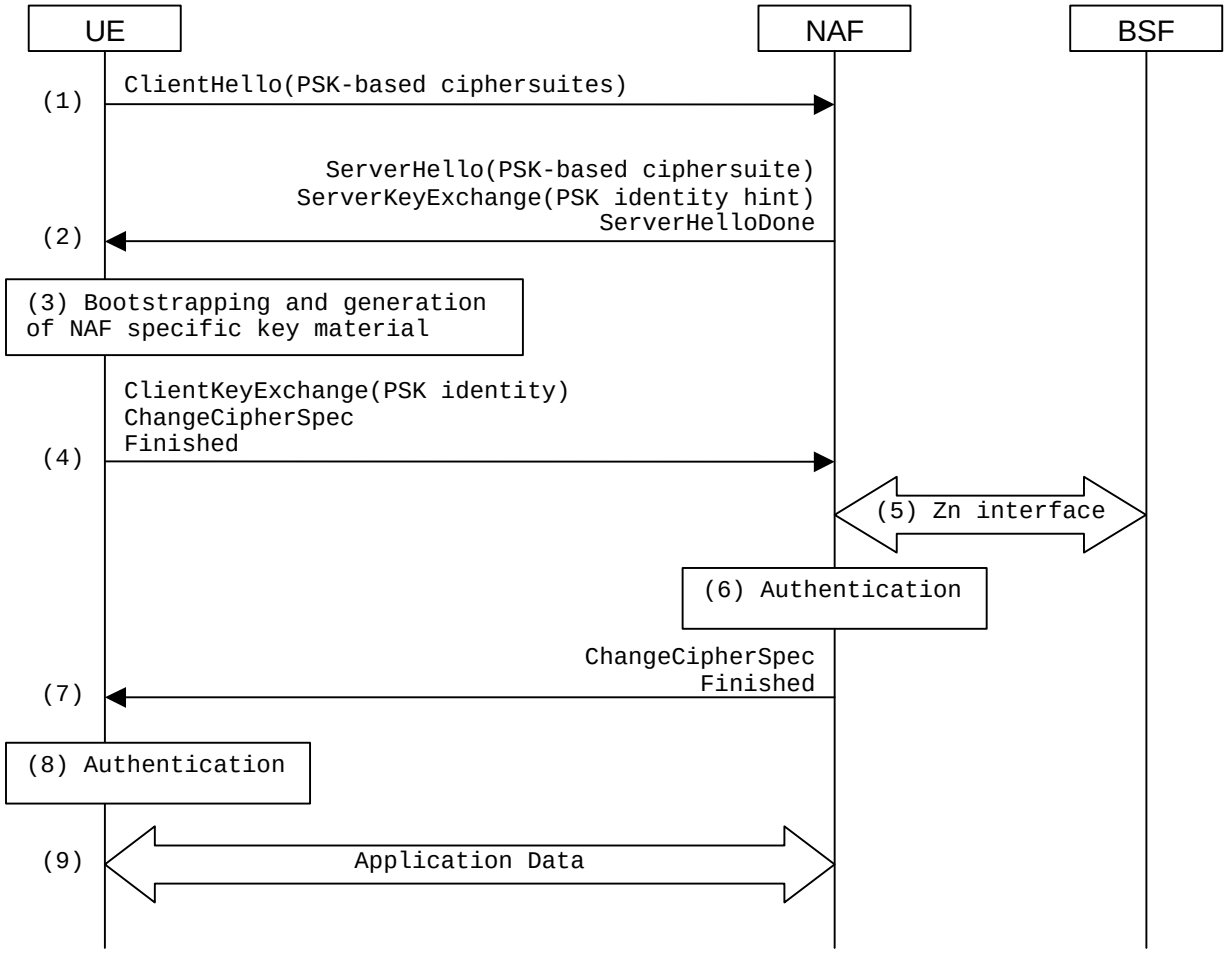

Figure F.3-1: PSK TLS handshake with bootstrapped security association.

1. **TLS handshake message: ClientHello (UE to NAF)**

   The UE sends ClientHello message to the NAF. In order to indicate that the UE is capable of PSK-based authentication it includes the PSK-based ciphersuites to the list of acceptable ciphersuites list. The UE also includes to the ClientHello message the server_name TLS extension containing the hostname of the NAF.

   TLS.ClientHello.client_version: the version of the TLS protocol in the UE shall be 3.1.

   TLS.ClientHello.random: a UE generated random structure.

   TLS.ClientHello.session_id: the ID of the TLS session is empty, i.e. no previous TLS session is used.

   TLS.ClientHello.cipher_suites: the list of ciphersuites shall include one or more PSK-based ciphersuites.

TLS.ClientHello.compression_methods: a list of the compression methods shall be null.

TLS.ClientHello.client_hello_extension_list: list of extensions shall include server_name extension that contains the hostname of the NAF.

Table F.3.1-1: Void

2. **TLS handshake messages: ServerHello, ServerKeyExchange, ServerHelloDone (NAF to UE)**

   If the NAF wants to use PSK-based authentication, it selects one of the acceptable PSK-based ciphersuites, places the selected ciphersuite in the ServerHello message, and includes an appropriate ServerKeyExchange message. The NAF may help the UE in selecting the correct PSK identity by providing a hint in ServerKeyExchange message. That hint could be, for example, 3GPP_bootstrapping@bsf.operator.com.

   TLS.ServerHello.server_version: the version of the TLS protocol in the NAF shall be 3.1.

   TLS.ServerHello.random: a NAF generated random (must be different from ClientHello.random.

   TLS.ServerHello.session_id: the identity of the TLS session generated by the BAF.

   TLS.ServerHello.cipher_suite: the ciphersuite selected by the NAF shall be one of the PSK-based ciphersuites listed in ClientHello.cipher_suites.

   TLS.ServerHello.compression_method: the compression method selected by the NAF shall be null.

   TLS.ServerHello.server_hello_extension_list: list of extensions shall be empty.

   TLS.ServerKeyExchange.psk_identity_hint: the PSK identity hint shall contain the constant string "3GPP-bootstrapping".

   TLS.ServerHelloDone: this message does not have data fields.

Table F.3.1-2: Void

3. **Bootstrapping and generation of NAF specific key material at UE**

   The UE performs the bootstrapping procedure to produce B-TID and Ks_NAF as described in clause A.3. If bootstrapping procedure has been done recently, the UE may use the B-TID and Ks_NAF produced from that procedure.

Table F.3.1-3: Void

4. **TLS handshake messages: ClientKeyExchange, ChangeCipherSpec, Finished (UE to NAF)**

   The UE sets the B-TID as the PSK identity, and Ks_NAF as the pre-shared key. The UE then sends ClientKeyExchange containing the B-TID, ChangeCipherSpec, and Finished messages to the NAF. The TLS premaster secret is derived from Ks_NAF as specified in draft-ietf-tls-psk-01 [14A].

   TLS.ClientKeyExchange.psk_identity: the PSK identity shall contain the B-TID.

   TLS.ChangeCipherSpec.type: contains value 1 (change_cipher_spec).

   TLS.Finished.verify_data: the verify data contains the hash of the handshake messages. For details, see RFC 2246 [10].

Table F.3.1-4: Void

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

   The NAF extracts the B-TID from the ClientKeyExchange message and requests the NAF specific key (Ks_NAF) from BSF. The BSF returns Ks_NAF that corresponds to the B-TID.

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

Table B.3.1-5: Bootstrapping authentication information procedure (NAF to BSF)

Message source and destination	Zn Information element name	Information Source in TLS	Description
NAF to BSF	B-TID	ClientKeyExchange.psk_identity	The bootstrapping transaction identifier is encoded in the ClientKeyExchange.psk_identity field according to PSK TLS.

6. **Authentication at NAF**

   The NAF validates the Finished message sent by the UE.

Table B.3.1-6: Void

7. **TLS handshake messages: ChangeCipherSpec, Finished (NAF to UE)**

   The NAF sends ChangeCipherSpec, and Finished messages to the UE.

   TLS.ChangeCipherSpec.type: contains value 1 (change_cipher_spec).

   TLS.Finished.verify_data: the verify data contains the hash of the handshake messages. For details, see RFC 2246 [10].

Table B.3.1-7: Void

8. **Authentication at UE**

   The UE validates the Finished message sent by the NAF.

Table B.3.1-8: Void

9. **Application data transfer**

   The UE and the NAF initiate application data transfer in the TLS session.

Table B.3.1-9: Void

Annex G (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
13/02/04		N1-040080			Version 0.0.1 Editor's internal draft		
13/02/04		N1-040173			TS skeleton		
13/02/04		N1-040174			Version 0.0.1 Editor's internal draft		
13/02/04		N1-040175			Version 0.0.1 Editor's internal draft		
13/02/04		N1-040176			Version 0.0.1 Editor's internal draft		
13/02/04		N1-040177			Version 0.0.1 Editor's internal draft		
19/03/04					Document updated with TS # 24.109		0.0.1
08/04/04		N1-040541			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
08/04/04		N1-040542			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
08/04/04		N1-040543			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
08/04/04		N1-040727			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
08/04/04		N1-040728			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
08/04/04		N1-040729			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
08/04/04		N1-040730			Version 0.1.0 incorporating results of CN1 discussions at CN1 #33bis (agreed documents N1-040541, N1-040542, N1-040543, N1-040727, N1-040728, N1-040729, N1-040730)	0.0.1	0.1.0
24/05/04		N1-040858			Version 0.2.0 incorporating the results of CN1 discussion at CN1#34 (agreed documents N1-040858, N1-041071, N1-041072)	0.1.1	0.2.0
24/05/04		N1-041071			Version 0.2.0 incorporating the results of CN1 discussion at CN1#34 (agreed documents N1-040858, N1-041071, N1-041072)	0.1.1	0.2.0
24/05/04		N1-041072			Version 0.2.0 incorporating the results of CN1 discussion at CN1#34 (agreed documents N1-040858, N1-041071, N1-041072)	0.1.1	0.2.0
21/06/04		N1-041166			Version 0.3.0 incorporating the results of CN1 discussion at CN1#34bis (agreed documents N1-041166, N1-041236, N1-041237, N1-041239, N1-041285, N1-041286)	0.2.0	0.3.0
21/06/04		N1-041236			Version 0.3.0 incorporating the results of CN1 discussion at CN1#34bis (agreed documents N1-041166, N1-041236, N1-041237, N1-041239, N1-041285, N1-041286)	0.2.0	0.3.0
21/06/04		N1-041237			Version 0.3.0 incorporating the results of CN1 discussion at CN1#34bis (agreed documents N1-041166, N1-041236, N1-041237, N1-041239, N1-041285, N1-041286)	0.2.0	0.3.0
21/06/04		N1-041239			Version 0.3.0 incorporating the results of CN1 discussion at CN1#34bis (agreed documents N1-041166, N1-041236, N1-041237, N1-041239, N1-041285, N1-041286)	0.2.0	0.3.0
21/06/04		N1-041285			Version 0.3.0 incorporating the results of CN1 discussion at CN1#34bis (agreed documents N1-041166, N1-041236, N1-041237, N1-041239, N1-041285, N1-041286)	0.2.0	0.3.0
21/06/04		N1-041286			Version 0.3.0 incorporating the results of CN1 discussion at CN1#34bis (agreed documents N1-041166, N1-041236, N1-041237, N1-041239, N1-041285, N1-041286)	0.2.0	0.3.0
26/08/04		N1-041419			Bootstrapping renegotiation indication in HTTP Digest	0.3.0	2.0.0
26/08/04		N1-041420			Key material delivery fix	0.3.0	2.0.0

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
26/08/04		N1-041423			Subscriber certificate enrolment to the main body	0.3.0	2.0.0
26/08/04		N1-041424			HTTP Digest: B-TID, and shared secret are ASCII based	0.3.0	2.0.0
26/08/04		N1-041428			Editorial fixes	0.3.0	2.0.0
26/08/04		N1-041594			Key to interpret HTTP signalling flows	0.3.0	2.0.0
26/08/04		N1-041595			Key to interpret TLS signalling flows	0.3.0	2.0.0
26/08/04		N1-041596			Subscriber authorization at PKI portal to obtain a particular type of certificate	0.3.0	2.0.0
26/08/04		N1-041597			Subscriber certificate enrolment with WIM authentication codes	0.3.0	2.0.0
26/08/04		N1-041598			Stage 3 for authentication proxy	0.3.0	2.0.0