3GPP TSG SA WG3 Security — S3#32
Edinburgh, UK, 9-13 February 2004

Tdoc ⌘S3-040073

CR-Form-v7

Pseudo CHANGE REQUEST

⌘ **33.221** **CR** **CRNum** ⌘**rev** **-** ⌘ Current version: **1.0.0** ⌘

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

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

Title:	⌘	Certificate chain content type
Source:	⌘	Nokia
Work item code:	⌘	GAA
Date:	⌘	30/1/2004
Category:	⌘	
Release:	⌘	Rel-6

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

Use one of the following releases:

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

Reason for change:	⌘	The format of certificate chain is not specified.
Summary of change:	⌘	The current TS does not specify the content-type for the certificate chain that is returned from the PKI portal. This pseudo CR adds the usage of certificate chain response type as it is specified in TLS Extensions specification (RFC 3546).
Consequences if not approved:	⌘	Certificate chain content-type is not specified.

Clauses affected:	⌘	2, 4.3.3.1.2.1, 4.4.1

Other specs affected: ⌘

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

Other comments: ⌘

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

***** BEGIN CHANGE *****

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*.

[<seq>] <doctype> <#>[([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

[1] "PKCS#10 v1.7: Certification Request Syntax Standard", RSA Laboratories, May 2000.

[2] Adams C., Farrell S., "Internet X.509 Public Key Infrastructure Certificate Management Protocols", RFC 2510, March 1999.

[3] Myers M., et al., "Internet X.509 Certificate Request Message Format", RFC 2511, March 1999.

[4] Chokhani S., et al, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 2527, March 1999.

[5] Franks J., et al, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[6] Housley R., et al, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[7] WAP-211-WAPCert, 22.5.2001: http://www1.wapforum.org/tech/terms.asp?doc=WAP-211-WAPCert-20010522-a.pdf

[8] WAP-260-WIM-20010712, 12.7.2001: http://www1.wapforum.org/tech/documents/WAP-260-WIM-20010712-a.pdf

[9] WAP-217-WPKI, 24.4.2001: http://www1.wapforum.org/tech/documents/WAP-217-WPKI-20010424-a.pdf

[10] ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1997, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", 1997.

[11] Draft 3GPP TS 33.220, "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture".

[12] Draft 3GPP TS 33.222, "Generic Authentication Architecture (GAA); Access to Network Application Function using HTTPS".

[13] Draft 3GPP TR 33.919, "Generic Authentication Architecture; System description".

[14] Open Mobile Alliance ECMA Crypto Library http://www.openmobilealliance.org

[15] Blake-Wilson, S., et al, "Transport Layer Security (TLS) Extensions", RFC 3546, June 2003.

***** END CHANGE *****

***** BEGIN CHANGE *****

4.3.3.1.2 Functionality and protocols

4.3.3.1.2.1 PKCS#10 with HTTP Digest Authentication

Editor’s note: This section uses HTTP Digest authentication to authenticate and integrity protect the certificate request and response. Shared key TLS is another solution to authenticate and protect the certificate enrolment, and whether it should be used instead of HTTP Digest is ffs.

HTTP Digest Authentication scheme [5] may be done with BSF shared key material the following way.

- UE makes a blank HTTP request to the NAF
- NAF returns a HTTP response with “WWW-Authenticate” header indicating that HTTP Digest Authentication is needed. Quality of protection (qop) attribute is set to “auth-int” meaning that the content in following HTTP requests and responses are integrity protected.
- UE calculates the correct response to the “WWW-Authenticate” header using the *identifier* (base64 encoded) as the username and the session key Ks (base64 encoded) as the password. The session key Ks is has been previously derived from the key material Ks that resulted from using Ub interface. HTTP Digest Authentication parameters are returned in the “Authorization” header of HTTP Response.
- NAF validates the “Authorization” header and upon successful validation, performs the requested task. In the corresponding HTTP response, NAF calculates the relevant values for “Authentication-Info” header, which is used to authenticate and integrity protect the NAF response.
- UE validates the “Authentication-Info” header and upon successful validation, accepts the payload in the HTTP response.

A PKCS#10 [1] based certification request is sent to the CA NAF using a HTTP POST request, which MUST be authenticated and integrity protected by HTTP Digest Authentication.

Certificate is delivered using the HTTP response, which MAY be authenticated and integrity protected by HTTP Digest Authentication. The content-type of the HTTP response depends on the response format. If a certificate is returned then it is “application/x-x509-user-cert”. If a pointer to the certificate is returned then it is “application/vnd.wap.cert-response” as specified in [9]. ~~The content-type and the format of the certificate chain is ffs.~~If a certificate chain is returned, then it is ”application/pkix-path” as specified in [15].

The UE requests a CA certificate delivery by sending a plain HTTP GET request with specific parameters in the request URI . The request MAY be authenticated and integrity protected by HTTP Digest Authentication.

CA certificate is delivered using the HTTP response, which MUST be authenticated and integrity protected by HTTP Digest Authentication. The content-type of the HTTP response would be “application/x-x509-ca-cert”. Note that the user should always be notified when a new CA certificate is taken into use.

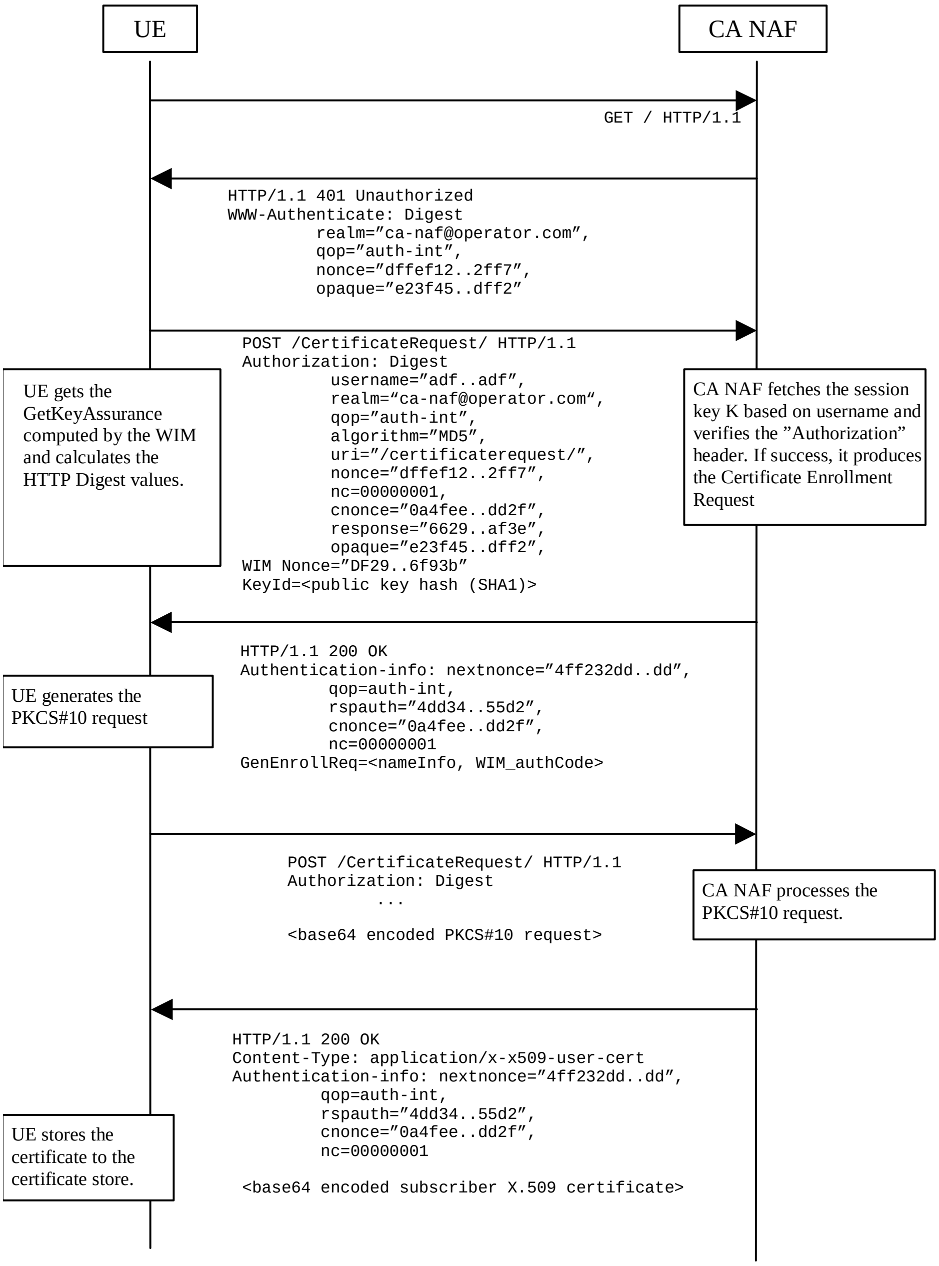
4.3.3.1.2.2 Key Generation

If the private key is stored in a UICC (e.g.in a WIM) and the UICC demands a special authorization (e.g. from the Operator) to generate the key, the ME may need to perform an HTTP POST request, which MAY be authenticated and integrity protected by HTTP Digest Authentication, to the NAF in order to deliver a nonce that is generated by the UICC. This will allow the NAF to authenticate directly to the UICC application and provide authorization for the key generation.

4.4 Certificate issuing procedure

Editor’s note: This section uses HTTP Digest authentication to authenticate and integrity protect the certificate request and response. Shared key TLS is another solution to authenticate and protect the certificate enrolment, and whether it should be used instead of HTTP Digest is ffs.

4.4.1 Certificate issuing



```
GET / HTTP/1.1
```

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
         realm=”ca-naf@operator.com”,
         qop=”auth-int”,
         nonce=”dffef12..2ff7”,
         opaque=”e23f45..dff2”
```

```
POST /CertificateRequest/ HTTP/1.1
Authorization: Digest
         username=”adf..adf”,
         realm=“ca-naf@operator.com“,
         qop=”auth-int”,
         algorithm=”MD5”,
         uri=”/certificaterequest/”,
         nonce=”dffef12..2ff7”,
         nc=00000001,
         cnonce=”0a4fee..dd2f”,
         response=”6629..af3e”,
         opaque=”e23f45..dff2”,
WIM Nonce=”DF29..6f93b”
KeyId=<public key hash (SHA1)>
```

UE gets the GetKeyAssurance computed by the WIM and calculates the HTTP Digest values.

CA NAF fetches the session key K based on username and verifies the ”Authorization” header. If success, it produces the Certificate Enrollment Request

```
HTTP/1.1 200 OK
Authentication-info: nextnonce=”4ff232dd..dd”,
         qop=auth-int,
         rspauth=”4dd34..55d2”,
         cnonce=”0a4fee..dd2f”,
         nc=00000001
GenEnrollReq=<nameInfo, WIM_authCode>
```

UE generates the PKCS#10 request

```
POST /CertificateRequest/ HTTP/1.1
Authorization: Digest
         ...

<base64 encoded PKCS#10 request>
```

CA NAF processes the PKCS#10 request.

```
HTTP/1.1 200 OK
Content-Type: application/x-x509-user-cert
Authentication-info: nextnonce=”4ff232dd..dd”,
         qop=auth-int,
         rspauth=”4dd34..55d2”,
         cnonce=”0a4fee..dd2f”,
         nc=00000001

 <base64 encoded subscriber X.509 certificate>
```

UE stores the certificate to the certificate store.

Figure 2: Certificate request using PKCS#10 with HTTP Digest Authentication.

The sequence diagram above describes the certificate request when using PKCS#10 with HTTP Digest. The sequence starts with an empty HTTP request to CA NAF. The CA NAF responds with HTTP response code 401 “Unauthorized” which contains a WWW-Authenticate header. The header instructs the UE to use HTTP Digest authentication.

The UE will generate the HTTP request by calculating the Authorization header values using the identifier it received from the BSF as username and the session key Ks. If the certificate request needs extra assurance by a WIM application for key Proof of Origin, the UE should include a WIM Nonce and the key id (i.e. SHA-1 public key hash) in this request

When CA NAF receives the request, it will verify the Authorization header by fetching the session key Ks from the bootstrapping server using the identifier, then calculating the corresponding digest values using K, and finally comparing the calculated values with the received values in the Authorization header. If the verification succeeds, the CA NAF may use the subscriber profile to compute and send back a GenEnrollReq attribute containing additional parameters that are needed for the following PKCS#10 request generation (e.g. nameInfo, WIM_authCode, ...). The CA NAF may use session key Ks to integrity protect and authenticate this response.

The UE will then generate the PKCS#10 request and send it to the CA NAF by using an HTTP Digest request. In the case that the private key is stored in a WIM application the ME should request the AssuranceInfo from the WIM application and include it in the PKCS#10 request, if provided (see annex B for all details). The AssuranceInfo provides a proof of origin for the key processing.(e.g. identifies the WIM application and provides a proof that the key is stored in it). UE may indicate the desired format of the certification response: a certificate, a pointer to the certificate (e.g., URL), or a full certificate chain (i.e., from the issued certificate to the corresponding root certificate). The enrolment request shall be as follows:

```
POST <base URL>?response=<indication>[other URL parameters] HTTP/1.1
Content-Type: application/x-pkcs10

<base64 encoded PKCS#10 blob>
```

where

<base URL> identifies a server/program.

<indication> used to indicate to the CA NAF what is desired response type for the UE. The possible values are: “single” for subscriber certificate only, “pointer” for pointer to the subscriber certificate, or “chain” for full certificate chain.

[other URL parameters] are additional, optional, URL parameters.

The incoming PKCS#10 request is taken in for further processing. If the CA NAF is actually a registration authority (RA NAF), the PKCS#10 request is forwarded to CA using any protocol available (e.g., CMC or CMP). After the PKCS#10 request has been processed and a certificate has been created, the new certificate is returned to the CA NAF. It will generate a HTTP response containing the certificate, or the pointer to the certificate as defined subclause 7.4 of [9], or a full certificate chain from issued certificate to the root certificate.

If the HTTP response contains the subscriber certificate itself, it shall be base64 encoded, and it may be demarcated as follows:

```
HTTP/1.1 200 OK
Content-Type: application/x-x509-user-cert

-----BEGIN CERTIFICATE-----
<base64 encoded X.509 certificate blob>
-----END CERTIFICATE-----
```

If the HTTP response contains the pointer to the certificate itself, the CertResponse structure defined in subclause 7.3.5 of [9] shall be used, and it may be demarcated as follows:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.wap.cert-response

-----BEGIN CERTIFICATE RESPONSE-----
<base64 encoded CertResponse structure blob>
-----END CERTIFICATE RESPONSE-----
```

If the HTTP response contains a full certificate chain in PkiPath structure as defined in [15] and~~, each certificate~~it shall be base64 encoded~~ and shall be demarcated as follows~~:

HTTP/1.1 200 OK
Content-Type: ~~ffs~~application/pkix-path

~~-----BEGIN CERTIFICATE-----~~
~~<base64 encoded X.509 certificate blob>~~
~~-----END CERTIFICATE-----~~
~~-----BEGIN CERTIFICATE-----~~
~~<base64 encoded X.509 certificate blob>~~
~~-----END CERTIFICATE-----~~
~~-----BEGIN CERTIFICATE-----~~
~~<base64 encoded X.509 certificate blob>~~
~~-----END CERTIFICATE-----~~

<base64 encoded PkiPath blob>

~~The certificates in the response are not needed to be in any particular order.~~ The content-type header value for the certificate chain is ~~ffs~~"application/pkix-path" as specified in [15].

The CA NAF may use session key Ks to integrity protect and authenticate the response, if a certificate or a pointer to the certificate is sent to the UE. The CA NAF shall use integrity protection and authenticate the response if full certificate chain is sent to the UE.

When UE receives the subscriber certificate, it is stored to local certificate management system.

NOTE: On board key generation is already defined in the WIM specification [8] issued by Open Mobile Alliance (OMA) group."

******* END CHANGE *******