**3GPP TSG SA WG3 Security — S3#31** **S3-030749**
**18 – 21 November, 2003, Munich, Germany**

*CR-Form-v7*

# Pseudo - CHANGE REQUEST

⌘ **ab.cde** **CR** **CRNum** ⌘**rev** **-** ⌘ Current version: 0.1.1 ⌘

*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:*** | ⌘ | Initial text for the TS |
| ***Source:*** | ⌘ | Siemens |
| ***Work item code:*** | ⌘ | GAA / HTTPS |
| ***Date:*** | ⌘ | 18 Nov 2003 |
| ***Category:*** | ⌘ | **B** |
| ***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 TS on GAA / HTTPS is currently void of content |
| ***Summary of change:*** | ⌘ | Text on the use of tunnels and authentication proxies is included |
| ***Consequences if not approved:*** | ⌘ | Risk of not completing TS on GAA / HTTPS in time |

| | | | | | |
|---|---|---|---|---|---|
| ***Clauses affected:*** | ⌘ | | | | |
| | | **Y** | **N** | | |
| ***Other specs affected:*** | ⌘ | **Y** | **N** | Other core specifications | ⌘ |
| | | | **N** | Test specifications | |
| | | | **N** | O&M Specifications | |
| ***Other comments:*** | ⌘ | | | | |

***** Begin of Change ****

# 4 Authentication Schemes

## 4.1 Requirements and principles

This document is based on the architecture specified in [TS33.220]. All notions not explained here can be found in [TS33.220].

*Editor's note: care must be taken that this specification is in line with TS 33.141 on presence security. SA3 has yet to decide the split between the two documents.*

## 4.2 Shared key-based UE authentication with certificate-based NAF authentication

This section explains how the procedures specified in [TS33.220] have to be enhanced when HTTPS is used between a UE and a NAF. The only enhancement required is the need to specify how the set up of a TLS tunnel is included in the general procedures specified in [TS33.220].

When the UE accesses a NAF, with which it does not yet share a key, then the sequence of events is as follows:

1. the UE runs http digest aka [rfc3310] with the BSF over the Ub interface.
2. If the BSF has no authentication vectors for the UE it fetches authentication vectors from the HSS over the Zh interface.

After the completion of step 1), the UE and the BSF share a secret key. This shared key is identified by a transaction identifier supplied by the BSF to the UE over the Ub interface key, cf. [TS 33.220, section 4.3.1].

~~2)~~3. The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate.

*Editor's note: TLS needs to be profiled in an appropriate section of this specification.*

~~3)~~4. The UE sends an http request to the NAF.

~~4)~~5. The NAF invokes http digest [rfc 2617] with the UE over the Ua interface in order to perform client authentication using the shared key agreed in step 1), as specified in [TS 33.220, Annex A].

~~5)~~6. While executing step 5), the NAF fetches the shared key from the BSF over the Zn interface, as specified in [TS 33.220, Annex A and section 4.3.2].

~~6)~~After the completion of step 4), UE and NAF are mutually authenticated as the TLS tunnel endpoints.

The UE may now run an appropriate application protocol with the NAF through the authenticated tunnel.

When the UE accesses a NAF, with which it already shares a key, steps 1), 5) and 6) may be omitted, as specified in [TS 33.220].

*Editor's note: the above procedure is generally applicable and conforms to [TS 33.220]. For the case of a co-located BSF and NAF an optimisation is possible which is currently located in the informative Annex Z. SA3 still needs to decide whether the material in the annex should be moved to the main body, or remain in an informative or normative annex, or be deleted.*

***** End of Change ****

***** Begin of Change ****

# 5 Use of authentication proxy

## 5.1 Requirements and principles

The authentication proxy may reside between the UE and the NAF as depicted in Figure y [*tba to section 5.2*]. The usefulness of an Authentication Proxy may be to reduce the consumption of authentication vectors and/or to minimize SQN synchronization failures.

The following requirements apply for the use of an Authentication Proxy:

- Authentication proxy shall be able to authenticate the UE using the means of Generic Bootstrapping Architecture, as specified in [Ts33.220].
- Authentication proxy shall send the authenticated identity of the UE to the application server belonging to the trust domain at the beginning of new HTTP session.
- Authentication proxy may not reveal the authenticated identity of the UE to the application server not belonging to the trust domain if required.
- The authenticated identity management mechanism shall not prevent the application server to use an appropriate session management mechanisms with the client.
- The UE shall be able to create multiple parallel HTTP sessions via the authentication proxy towards different application servers.

NOTE: The used session management mechanism is out of the scope of 3GPP specifications.

## 5.2 Authentication proxy architecture

<include figure y here>

The use of an authentication proxy (AP) is fully compatible with the architecture specified in [TS33.220] and in section 4 of this specification. When an AP is used in this architecture, the AP takes the role of a NAF. When an https request is destined towards an application server behind an authentication proxy (AP), the AP terminates the TLS tunnel and performs UE authentication. The AP proxies the http request to the application server.

*Editor's note: if contribution S3-030xxx to SA3#31 on " Technical solutions for access to application servers via Authentication Proxy and HTTPS" is agreed, an annex is added, containing informative material on authentication proxies. Section 5.2. should then contain a reference to this annex.*

***** End of Change ****

***** Begin of Change ****

# Annex Z: OPTIMISED SEQUENCE OF EVENTS FOR ACCESS TO CO-LOCATED BSF AND NAF VIA HTTPS

*Editor's note: SA3 still needs to decide whether the material in the annex should be moved to the main body, or remain in an informative or normative annex, or be deleted.*

*Editor's note: the material in this annex is based on the information flow in S3-030371, Annex A.*

When the UE accesses a NAF, and the NAF is co-located with the BSF, then the optimised sequence of events is as follows:

1. The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate.

*Editor's note: TLS needs to be profiled in an appropriate section of this specification.*

2. If the UE does not share a key with the NAF, the UE sends an http request to a NAF, containing the UE's identity.

3. If the NAF receives an http request from the UE without an Authorization header, or with an Authorization header it does not accept, the NAF contacts the (co-located) BSF to obtain a challenge and a password, computed from an AKA authentication vector according to [draft-torvinen-http-digest-aka-v2].

4. If the BSF has no authentication vectors for the UE it fetches authentication vectors from the HSS over the Zh interface.

5. The NAF replies to the UE by sending a 401 "unauthorized" message with a WWW-Authenticate header according to [draft-torvinen-http-digest-aka-v2].

6. The UE sends an http request to the NAF with an Authorization header according to [draft-torvinen-http-digest-aka-v2].

7. The NAF verifies the Authorization header.

After the completion of step 7), UE and NAF are mutually authenticated as the TLS tunnel endpoints.

8. The NAF replies to the http request returning the requested information to the UE, if any.

The UE may now run an appropriate application protocol with the NAF through the authenticated tunnel.

*Editor's note: the transport of of key derivation information from NAF/BSF to UE needs further study.*

***Note on co-location of BSF and NAF:*** *a BSF and a NAF may be combined on one machine in such a way that the BSF is accessed through http, not using TLS, and the NAF is accessed through https. From a functional point of view, this case is identical to the general case described in section 4.2. It is even possible to functionally duplicate the BSF on one machine in such a way that the BSF is accessed through http, when TLS is not required, and accessed through https, when access to the NAF requires TLS.*

***Editor's note on carrying identities****: the first http request after TLS set-up needs to contain the identity of the UE. The reason is that for http digest the server can issue a challenge without knowing the client's identity, whereas for http digest aka the challenge is specific to a particular client. There seem to be at least two solutions for this:*
*a) use a specially formed http GET request, as described for the Ub interface in [TS33.220].*
*b) use an Authorization header with dummy values (to be defined). The server will not accept the credentials, and will reply with a 401 "unauthorised". For maximum harmonisation, the UE identity, which needs to be included by the UE at the start of the http digest aka protocol run, should be carried in the same way in the general and the optimised case.*

***Note on tunnelled authentication and the use of http digest aka:***
*In this annex and in section 4.2 respectively, different versions of http digest aka are used. This prevents man-in-the-middle attacks with tunnelled authentication. Version 1 of http digest aka [rfc 3310] is used between the UE and the BSF when http digest aka is NOT used to authenticate the client endpoint of a TLS tunnel extending between UE and BSF. Version 1 may be run inside or outside a TLS tunnel, as long as it is not used for client authentication. Version 2 [draft-torvinen-http-digest-aka-v2] is used when http digest aka IS used to authenticate the client endpoint of a TLS tunnel. Version 2 is always run inside a TLS tunnel.*

***Editor'Note on tunnelled authentication and the use of http digest aka****:*
*Instead of using different versions of http digest aka to distinguish whether http digest aka is used for client authentication of a TLS tunnel or not, this distinction could be provided by different means. Possibilities suggested on the SA3 mailing list include to extend the specification of http digest akav2 to include a "situation" (or "context") parameter in the computation of the password, then always use http digest akav2, but with different values for the "situation" parameter for the two different uses.*

***Note on transaction identifiers****: the general approach, as specified in section 4, which is based on [TS 33.220], requires the use of a transaction identifier over the interfaces Ua, Ub and Zn. The use of such a transaction identifier is neither possible nor necessary in the optimised case described in this annex*

***** End of Change ****