3GPP TSG SA WG3 Security — S3#29 **S3-030341**
15 –18 July 2003
San Francisco, USA

Source: **Nokia**

Title: **Transaction identifier location in protocol A**

Document for: **Discussion and Decision**

Agenda Item: **7.9 (Support for subscriber certificates)**

1. Introduction

This contribution concerns one issue in the attached pseudo CR for Protocol A, namely how the transaction identifier (TID) is transferred from BSF to UE in protocol A. Note this discussion affects only the step 11 in the attached pseudo CR.

In order to secure comminication between UE and NAF with key material bootstrapped from AKA, transaction identifier (TID) must be transferred from BSF to UE in the last HTTP response (200 OK) message (see [TS-SSC]). This contribution lists two possible alternatives to transfer the TID: as a Content-Location header in the HTTP response headers and as an XML document in the HTTP response payload.

2. Discussion

2.1 HTTP Header: Content-Location

TID can be transferred using Content-Location header in the HTTP response. For example:

```
HTTP/1.1 200 OK
Content-Location: /tid/123456
Content-Length: 0
Authentication-Info:
  qop=”auth”,
  rspauth="6629fae49393a05397450978507c4ef1",
  cnonce="0a4f113b"
```

The value of the Content-Location header value would be used by UE when it is accessing NAF and NAF would retrieve the session key (and possible additional subscriber profile information) from BSF using this value.

Pros and cons:

+ simple
- headers are not integrity protected by HTTP Digest (hence qop=”auth”; server-authenticated TLS tunnel can be used to integrity protect the whole message)

2.2 XML document in HTTP response payload

TID can be transferred using HTTP Response payload, which would contain an XML document containing the TID. For example:

```
HTTP/1.1 200 OK
Authentication-Info:
  qop=”auth-int”,
  rspauth="6629fae49393a05397450978507c4ef1",
  cnonce="0a4f113b"
Content-Type: application/3gpp-bsf+xml
Content-Length:

<?xml version="1.0" encoding="UTF-8"?>
<bsf xmlns="urn-to-xml-schema-of-3gpp-bsf"
    bsf-tid="base64 encoded TID"/>
```

Where the XML Schema definition could be:

```
<?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="bsf-tid" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>

</xs:schema>
```

The value of the attribute “bsf-tid” in “bsf” element would be used by UE when it is accessing NAF and NAF would retrieve the session key (and possible additional subscriber profile information) from BSF using this “bsf-tid” attribute value.

Pros and cons:

+ payload can be integrity protected by HTTP Digest AKA (hence qop=”auth-int”)
- content-type “application/3gpp-bsf+xml” must be registered by IANA

3. Proposal

We propose in the attached pseudo CR to use XML document in the HTTP response payload (section 2.2) to transfer the TID from BSF to UE because then the payload is integrity protected using HTTP Digest Authentication.

4. References

[RFC3310] A. Niemi, et al, “Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)”, RFC3310, September 2002.

[TS-SSC] Bootstrapping of application security using AKA and Support for Subscriber Certificates; System Description (Release 6), V0.2.0.

CR-Form-v7

CHANGE REQUEST

⌘ **SpecNumber** CR **CRNum** ⌘ **rev** **-** ⌘ Current version: **0.2.0** ⌘

*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 [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Protocol A in stage 3 detail
Source:	⌘	Nokia
Work item code:	⌘	SEC1-SC
Date:	⌘	30/06/2003
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:	⌘	Current specification TS does not contain the protocol transaction in stage 3 detail.
Summary of change:	⌘	Protocol A is given in stage 3 detail, including both successful and failure cases.
Consequences if not approved:	⌘	Implementation detail is missing.

Clauses affected:	⌘	2, 4.3			
Other specs affected:	⌘	**Y**	**N**		
			X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

2 References

The following documents contain provisions that, 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] 3GPP TR 41.001: "GSM Release specifications".

[2] 3GPP TR 21.912 (V3.1.0): "Example 2, using fixed text".

[3] 3GPP TS 31.102: "Characteristics of the USIM Application".

[4] 3GPP TS 33.102: "Security Architecture".

[5] 3GPP TS 23.003: "Numbering, addressing and identification".

[PKCS10] "PKCS#10 v1.7: Certification Request Syntax Standard", RSA Laboratories, May 2000.

[RFC2510] Adams C., Farrell S., "Internet X.509 Public Key Infrastructure Certificate Management Protocols", RFC 2510, March 1999.

[RFC2511] Myers M., et al., "Internet X.509 Certificate Request Message Format", RFC 2511, March 1999.

[RFC2527] Chokhani S., et al, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework", RFC 2527, March 1999.

[RFC2617] Franks J., et al, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC 3023] M. Murata, et al, "XML Media Types", RFC 3023, January 2001.

[RFC3280] Housley R., et al, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[RFC 3310] A. Niemi, et al, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC3310, September 2002.

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

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

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

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

********** NEXT CHANGE **********

4.3 Procedures

This chapter 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 latter the key material generation procedure.

4.3.1 Bootstrapping procedures

The bootstrapping procedures consist of executions of the protocol A and protocol C that are specified in stage 3 detail in sections 4.3.1.1 and 4.3.1.2 respectively. The following overall message sequence diagram outlines the whole successful bootstrapping procedure.

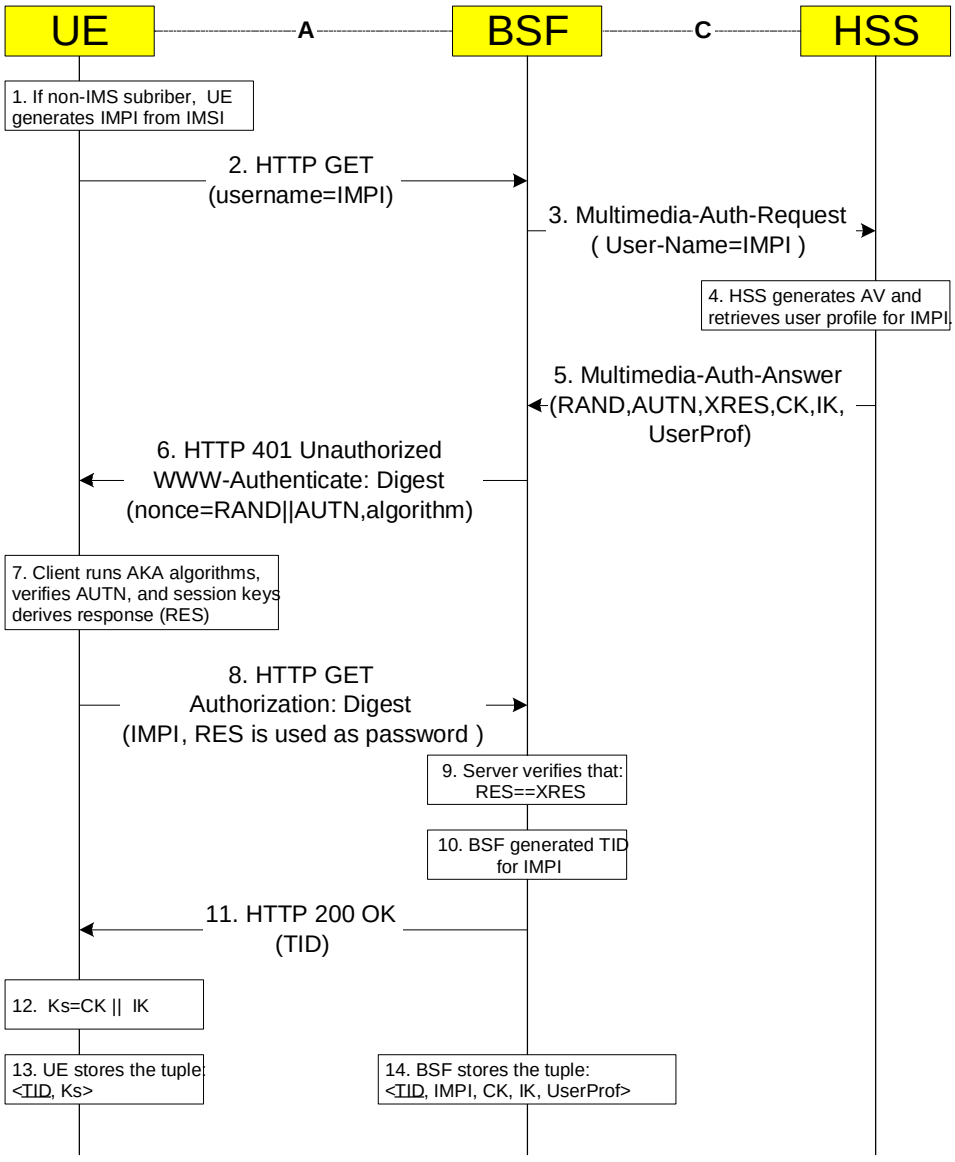

Figure 3: The bootstrapping procedure

Steps 1-2 and 6-14 are described in the A interface chapter 4.3.1.1 and steps 3-5 in the B interface chapter 4.3.1.2.

4.3.1.1 Protocol A

4.3.1.1.1 Successful case

When a UE wants to interact with an NAF, it shall first perform a bootstrapping authentication (see Figure 3).

~~*:Editor's notes: Protocol C related procedure will be added here in future development. It may re-use Cx interface that is specified in TS 29.228.*~~

1~~: The UE sends an HTTP request towards the BSF~~. A non-IMS subscriber shall construct a private user identity from IMSI number according to [5].

2.The UE sends an HTTP request containing the User Private Identity (IMPI) towards its home BSF. The IMPI is given in Request-Line.

```
GET /?username=impi HTTP/1.1
```

~~2~~3-5. BSF retrieves the user profile and a challenge, i.e. the Authentication Vector (AV)~~, AV = RAND||AUTN||XRES||CK||IK)~~ by protocol C from the HSS (c.f. section 4.3.1.2).

~~3~~6. Then BSF forwards the challenge to the UE in the HTTP 401 Unauthorized message (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 [RFC3310].

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
  realm="bsfServer@operatornetwork",
  nonce="CjPk9mRqNuT25eRkajM09uTl9nM09uTl9nMz5OX25PZz==",
  qop=”auth-int”,
  opaque="5ccc069c403ebaf9f0171e9517f40e41",
  algorithm=AKAv1-MD5
```

~~4:~~7. The UE calculates the message authentication code (MAC) so as to verify the challenge from authenticated network; the UE also calculates CK, IK and RES. This will result in session keys IK and CK in both BSF and UE.

~~5~~8. The UE sends request again, with the Digest AKA RES as the response to the BSF.

```
GET /?username=impi HTTP/1.1
Authorization: Digest
  username="impi",
  realm=" bsfServer@operatornetwork",
  nonce="CjPk9mRqNuT25eRkajM09uTl9nM09uTl9nMz5OX25PZz==",
  uri="/?username=impi",
  qop=”auth-int”,
  nc=00000001,
  cnonce="0a4f113b",
  response="6629fae49393a05397450978507c4ef1",
  opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

~~6:~~9. The BSF server shall check the received Digest equals to the expect value. If so, it means~~If~~ the RES equals to the XRES that is in the AV~~,~~ which means the UE is authenticated.

10. The BSF generates a TID for the IMPI.

~~7~~11. The BSF shall send the successful response in 200 OK message to the UE to indicate the success of the authentication.

The BSF shall insert the transaction ID into the message body that is integrity-protected, and the calculation of response digest is specified in [RFC2617]. The multipart definition follows [RFC 3023].

```
HTTP/1.1 200 OK
Authentication-Info:
  qop=”auth-int”,
  rspauth="6629fae49393a05397450978507c4ef1",
  cnonce="0a4f113b"
Content-Type: application/3gpp-bsf+xml
Content-Length:

<?xml version="1.0" encoding="UTF-8"?>
<bsf xmlns="urn-to-xml-schema-of-3gpp-bsf "
    bsf-tid="base64 encoded TID"/>
```

The XML Schema of the bsf element is specified below. The uniform resource name (URN) for 3gpp-bsf namespace is FFS.

```
<?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="bsf-tid" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

~~8~~12. The key material Ks is generated in ~~both BSF and~~ UE by concatenating CK and IK. The Ks is used for securing the protocol B.

Editor’s note: The key material Ks is 256 bits long. It is up each NAF to make the usage of the key material specifically.

13. The UE stores the tuple <TID,Ks>

14. The BSF stores the tuple <TID,IMPI,CK,IK,UserProfile>

~~9. BSF may supply a transaction identifier to UE in the cause of protocol A.~~

4.3.1.1.2 User authentication failure case

If the response is verified to be different than expected, the BSF shall send a HTTP 401 Unauthorized message in step 8, indicating that BSF does not wish to accept the request. It may return a HTTP 401 Unauthorized response that includes a WWW-Authenticate header field containing another challenge applicable to the requested resource.

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
  realm="bsfServer@operatornetwork",
  qop="auth-int",
  nonce="9uQzNPbk9jM05Pbl5Pbl5DIz9uTl9uTl9jM0NTHk9uXk==",
  opaque="dcd98b7102dd2f0e8b11d0f600bfb0c093",
  algorithm=AKAv1-MD5
```

After N failed attempts of authentication procedure, the application may indicate end user a failure message. The exact value of N is defined by local policy.

4.3.1.1.3 Network authentication failure case

In case the UE fails at authenticating the network based on the MAC generated locally, the UE shall abort the procedure from step 6.

4.3.1.1.4 Synchronisation failure

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

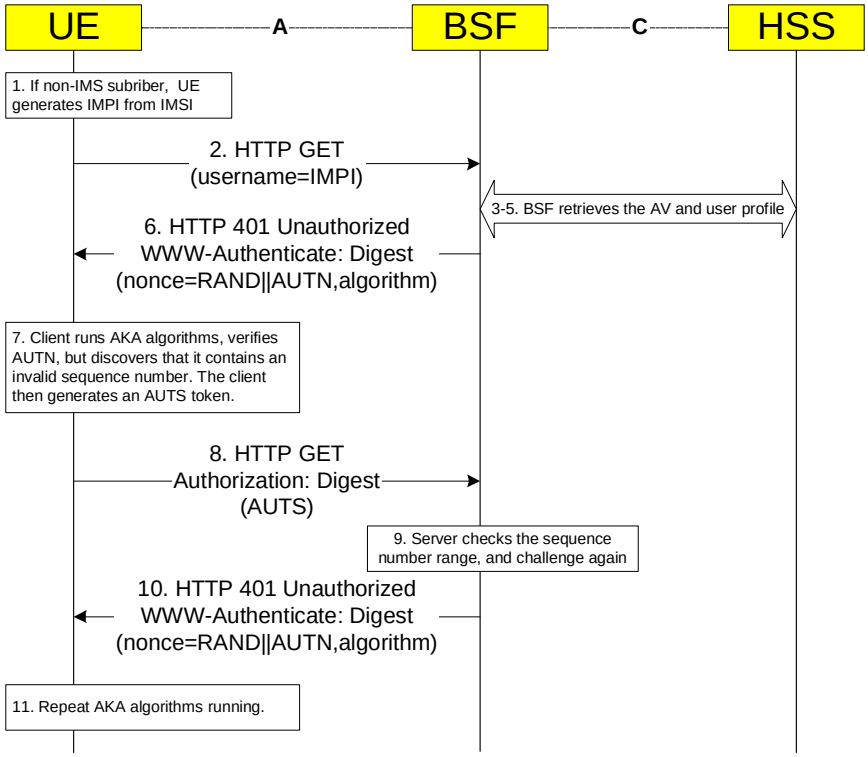

Figure 4: The message flow of bootstrapping procedure in sequence number synchronization failure case

The message flow in Figure 4 shall be different since step 7 in the Figure 3. The client identifies the sequence number is out of synchronization. The client shall generate the AUTS parameter according to [4].

In step 8, AUTS parameter is populated in Authorization header, as specified in [RFC3310].

```
GET /?username=impi HTTP/1.1
Authorization: Digest
  username="impi",
  realm="bsfServer@operatornetwork",
  nonce="CjPk9mRqNuT25eRkajM09uTl9nM09uTl9nMz5OX25PZz==",
  uri= ”/?username=impi”,
  qop=”auth-int”,
  nc=00000001,
  cnonce="0a4f113b",
  response="4429ffe49393c02397450934607c4ef1",
  opaque="5ccc069c403ebaf9f0171e9517f40e41",
  auts="5PYxMuX2NOT2NeQ="
```

In step 10, the BSF shall send another challenge based on new range of sequence number.

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
  realm="bsfServer@operatornetwork",
  qop="auth-int",
  nonce="9uQzNPbk9jM05Pbl5Pbl5DIz9uTl9uTl9jM0NTHk9uXk==",
  opaque="dcd98b7102dd2f0e8b11d0f600bfb0c093",
  algorithm=AKAv1-MD5
```