3GPP TSG SA WG3 Security#30 **S3-030551**
6 – 10 October, 2003
Povoa do Varzim, Portugal

Agenda Item: **7.9 (GAA)**

Source: **Siemens**

Title: **Generic Bootstrapping Architecture evaluation**

Document for: **Discussion and Decision**

1. Introduction

At the 3GPP SA3#29 meeting it was decided that SA3 will attempt to provide one authentication solution suitable for many Rel-6 applications (MBMS, presence,...). This work was initiated while different authentication proposals were on the table, each specifically tailored to the application and having different shortcomings. The cost of authentication vectors, a multiplicity of interfaces to the HSS and authentication synchronisation failures were among the most cited ones.

The work has been progressed at the SA3 Adhoc meeting in Antwerp and has resulted in a tentative solution for a Generic Bootstrapping[1]Architecture (GBA) and a list of requirements on a Generic Authentication Architecture (GAA). The GBA contains the core functionality to provide the different Rel-6 (and future applications) with shared secrets. The shared secrets are bootstrapped from the security association shared between USIM and HSS. This security association is part of a 3G subscription.

Regarding the GAA work it then remains to be decided by SA3 for every application in Rel-6 whether shared secrets, provided by means of the GBA, or subscriber certificates, or a third solution shall be used. Subscriber certificates may be obtained by UEs based on the bootstrapped shared secrets from the GBA, using the procedures specified in the draft TS 33.109 on "Bootstrapping of application security using AKA and support for subscriber certificates", or may be obtained by other means. (The latest version can be found in S3-030488). SA3 may also work on how to apply the GBA to features defined outside 3GPP (e.g. by OMA) in the future. Discussion on this, however, has not yet started due to lack of candidate features from outside 3GPP, presented to SA3.

This contribution continues the evaluation of alternatives for a GBA started in the Siemens contribution S3z030011 to the SA3 Adhoc in Antwerp.

2. State of discussion

The GBA work will build on the contents of section 4 of the draft TS 33.109. The specification of the GBA is to be contained in this TS, unless SA3 decides to create a new TS for this purpose.

Within the framework of draft TS 33.109, the GBA is to provide shared secrets to a UE and a Network Application Function (NAF). These shared secrets are bootstrapped from the security association shared between USIM and HSS, which comes with a 3G subscription. A Bootstrapping Server Function (BSF) interfaces with the HSS, the UE, and the NAF. An overview of the architecture of a GBA is shown in the following figure 1 taken from draft TS 33.109.

[1] The term "bootstrapping", as used by 3GPP SA3, refers to a procedure, or procedures, which allow to derive short-term cryptographic keys (shared secrets) from the long-term 3G AKA authentication key. This long-term key comes with a 3G subscription and is stored in the USIM and in the Authentication Centre in the HSS. In this way, the bootstrapping procedure enables 3G subscribers and servers, which may have never had contact before, to securely communicate.

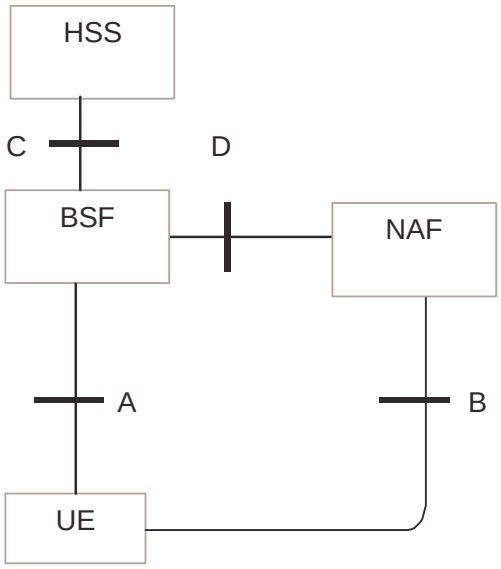

Figure 1: Simple network model for bootstrapping

Protocol A is realised in draft TS 33.109 through http digest aka (rfc 3310). Stage 3 details are provided in S3-030341 (submitted to SA3#29). In a variant with tunnelled authentication, it may be advisable to use http digest akav2 instead (see section 3.2 below). Protocol B is application-dependent. TDs S3-030342 and S3-030343 (submitted to SA3#29) provides stage-3 details on protocols C and D. The protocols C and D will be DIAMETER-based and may build on the Cx-interface defined for the IMS system, between S-CSCF and HSS. With this architectural approach, an implementation of the BSF will become relatively efficient, as already existing components of the IMS (Cx interface, HTTP-Digest as authentication protocol) can be re-used for the GBA.

Use of authentication proxies (cf. S3z030011, section 3, or S3-030371):

When http servers are accessed through https (http over TLS) it may be useful to employ reverse http authentication proxies. Such an authentication proxy would terminate the TLS tunnel from the UE for multiple http-based application servers sitting behind the proxy, and would take care of UE authentication. It was agreed at SA3#29 that proxy functionality providing a termination point for TLS should be used. It was expected that the HTTP authentication proxy would be able to provide single sign on towards multiple 3GPP HTTP based application servers, by reducing the required TLS connections between UE and proxy to only one. However, a companion contribution by Siemens to this meeting shows that this assumption may be problematic. But other advantages of a reverse authentication proxy may still remain. Therefore a GBA shall be compatible with the use of reverse http authentication proxies.

The following list provides an overview of the **status of the work for the different applications** under the scope of the GAA.

- **HTTP based administration via the Ut reference point:** over the Ut reference point, a user can manage his data on an application server (e.g. presence server, conferencing server). The application server, or an authentication proxy in front of the application server, would be the NAF. TLS would be used to authenticate the server to the UE and protect the http payload between the NAF and the UE. UE authentication would be provided through a variant of http digest, based on the shared secret provided by the BSF.

- **MBMS key distribution:** if the GBA was used for MBMS key distribution, the BM-SC would play the role of a NAF. Protocol B would consists in the delivery of a key encryption key from BM-SC to UE. Protocol B would include mutual authentication between the UE and the BM-SC. UE authentication, and BM-SC authentication, would be based on the shared secret provided by the BSF.
  SA3 has not yet decided whether the distribution of an MBMS key encryption key will run over http or not. The Siemens Contribution S3z030010, presented at the SA3 adhoc meeting, pointed out that the use of authentication proxies for MBMS key distribution, as described in S3-030367, would have several disadvantages. But if authentication proxies were used they could play the role of NAF instead of the BM-SC.
- **Support for subscriber certificates:** in this case, the NAF would be a PKI portal, and protocol B would be a certificate enrolment protocol, e.g. CMP, or PKCS#10 with http digest authentication. For details see draft TS 33.109.
- **3G-WLAN interworking - UE initiated End to End tunneling:** The problem to be solved here is how to set up a secure tunnel. No contributions were discussed in SA3 so far as SA2 only recently decided on the tunnelling concept. (But there is a Siemens contribution on this issue to this meeting.) If the GBA was applied to this feature the NAF would be the PDG terminating the UE-initiated tunnel on the network side.
- **GUP access:** The use case (see S3-030316 for GUP use cases) where a GUP-client on the UE wants to access data via a GUP-server (the NAF), might be suitable for using the GBA. An analysis has still to be conducted by SA3.
- **OMA applications:** This discussion may be started after the GAA guidelines and the GBA architecture have been matured further.

The following application, as currently specified in TS 23.234 and TS 33.234, cannot use the GBA.

- **3G-WLAN interworking - network access:** the proposed solution is the use of EAP-SIM or EAP-AKA within the IEEE 802.1X framework, cf. draft TS 33.234, latest version in S3-030492. The GBA work relies on an http connection between the UE and the BSF which is not available at the time the EAP-run takes place. Therefore, the GBA does not apply.

3. Architectural alternatives

3.1 General approach using bootstrapping of application security based on draft TS 33.109

The general approach to bootstrapping of application security presented in this subsection is compatible with draft TS 33.109.

However, draft TS 33.109 does not address the question of tunnelled authentication. Therefore, draft TS 33.109 needs to be extended to take tunnelled authentication into account. It is further proposed that different tunnelled authentication methods are explicitly addressed in the specifications, rather than addressing tunnelled authentication in a generic fashion. The reason for this proposal is to minimise the risk of the well-known man-in-the-middle attacks through flawed procedures. The only tunnelled authentication method currently under discussion in SA3 is the access to http servers via https (http over TLS). In this method, server authentication is provided through TLS by means of server certificates while client authentication is performed at the application layer, using a variant of http digest.

The general approach is also compatible with the use of a reverse http authentication proxy (AP). From a GBA point of view, an AP is just another example of a NAF as defined in TS 33.109.

Due to the special treatment required for tunnelled authentication, the following figure 2 distinguishes between the case where NAFs are accessed via https, and the general case[2]. In general, NAFs will use different protocols for protocol B, depending on the application. Examples of protocol B are contained in draft TS 33.109 for the case the NAF is a portal for supporting subscriber certificates. A generic example for an http-based protocol B is also contained in draft TS 33.109.

It is outside the scope of this contribution how the UE can discover a BSF. This seems rather a task for SA2 or CN1 to solve. Once the GBA architecture is reasonably stable, SA2 and CN1 should be informed of this issue.

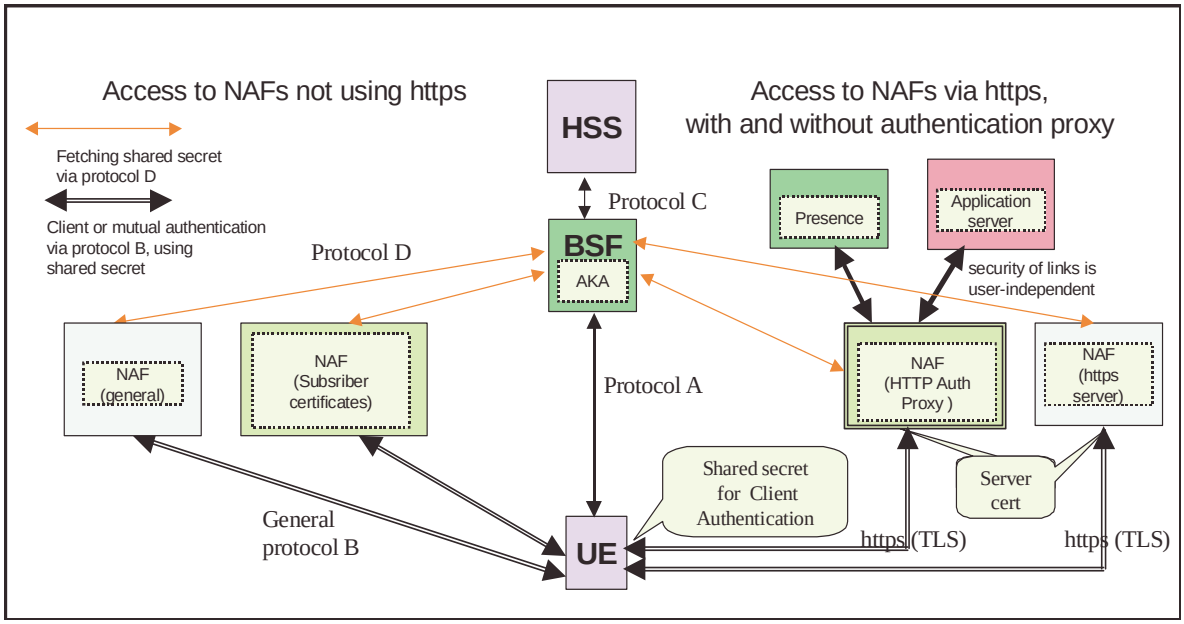

Figure 2: general approach to bootstrapping of application security

Case 1: Access to NAFs not using https

This case is identical to what is currently specified in draft TS 33.109. No additions to draft TS 33.109 are required as no TLS tunnels are involved here.

When the UE wants to access one of the application servers (which may be HTTP-based or not) on the left hand side of figure 2, then the sequence of events is as follows (overview):

1) the UE starts http digest aka (rfc3310, protocol A) with the BSF. The BSF may contact the HSS to fetch authentication vectors (protocol C). After step 1), the UE and the BSF share a secret key, cf. TS 33.109, section 4.3.1.

2) The UE sends a request (e.g. an http request) towards the application server (NAF).

3) The UE runs protocol B with the NAF using the key agreed in step 1) (e.g. http digest = RFC2617 to perform client authentication, as described in S3-030357). In the process, the NAF fetches the agreed key from the BSF (protocol D), as described in draft TS 33.109, section 4.3.2.

4) The UE runs the application protocol with the NAF.

---

[2] Figure 2 is an update of the figure in section 6.1 of S3z030011. There, a distinction was made between http based services and other services. The distinction made between https-based services and other services seems more to the point, as http-based services not using TLS can be treated in the same way as general services.

Case 2: Access to NAFs via https

When the UE wants to access one of the application servers on the right hand side of figure 2, then the sequence of events is as follows (overview):

1) the UE starts http digest aka (rfc3310, protocol A) with the BSF. The BSF may contact the HSS to fetch authentication vectors (protocol C). After step 1), the UE and the BSF share a secret key, cf. TS 33.109, section 4.3.1.

2) The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a server certificate.

3) The UE sends an https request towards a NAF.

(From a UE point of view, in order to perform steps 2) and 3), the browser just sends an https request to https:// ...NAF.com.)

4) The UE runs http digest (rfc 2617, protocol B) with the NAF to perform client authentication using the key agreed in step 1), as described in S3-030357 (a pseudo-CR to TS 33.109 presented at SA3#29). In the process, the NAF fetches the agreed key from the BSF (protocol D), as described in TS 33.109, section 4.3.2.

5) The UE runs the application protocol with the NAF.

This sequence of events conforms to TS 33.109. But the set up of the TLS tunnel, which is not part of TS 33.109, had to be added here. As long as the BSF and the NAF are distinct entities, this sequence of events is fine. But when they are co-located then this sequence is no longer optimally efficient (cf. subsection 3.2).

Consideration on authentication proxies

When an https request is destined towards an application server which sits behind an reverse authentication proxy (AP), the AP completes TLS tunnel set-up and client authentication with the UE. The AP proxies the http request to the application server. I.e. steps 2) to 4) above are performed with the AP playing the role of a NAF. The UE is, in general, not aware that its requests are proxied by an AP.

Proposal: *it is proposed that the procedures described in this subsection 3.1 become part of a GBA. Provided this is agreed by SA3 then pseudo-CRs have to be written against section 4 of TS 33.109 on the following two issues:*

- *take into account TLS tunnel set-up in the procedures in TS 33.109;*
- *include the above consideration that a NAF can also be an authentication proxy.*

As far as can be seen, no further technical issues need to be solved regarding the contents of this subsection. So, the current text of section 4 of draft TS 33.109, together with the two pseudo CRs on the above issues added, would constitute a workable GBA. In section 3.2 it is discussed whether this text should be enhanced to provide optimisations for certain configurations.

3.2 Approach using bootstrapping of application security with co-located BSF and NAF.

This subsection shows a possibility for optimisation of case 2 of subsection 3.1 (access via https) in the special case of co-located BSF and NAF. If it is decided to standardise this optimisation then it is proposed that the corresponding specification would supplement rather than replace the general approach from subsection 3.1. I.e., instead of using the sequence of events described in this subsection, the sequence of events from subsection 3.1, case 2, could also be applied to this special case.

The co-location of BSF and NAF is a design alternative, which allows to reduce the number of different pieces of equipment. An additional advantage would be that protocol D between BSF and NAF would not have to be implemented. Figure 3 below shows the case where the NAF is an authentication proxy (AP), and application servers are sitting behind the NAF. The NAF in figure 3 could also be just an application server. In that case, no further application servers would be sitting behind the NAF.

Note: it needs to be investigated further whether a physical separation of the authentication and key agreement functionality (BSF) from the application traffic would be desirable because it would provide enhanced security.

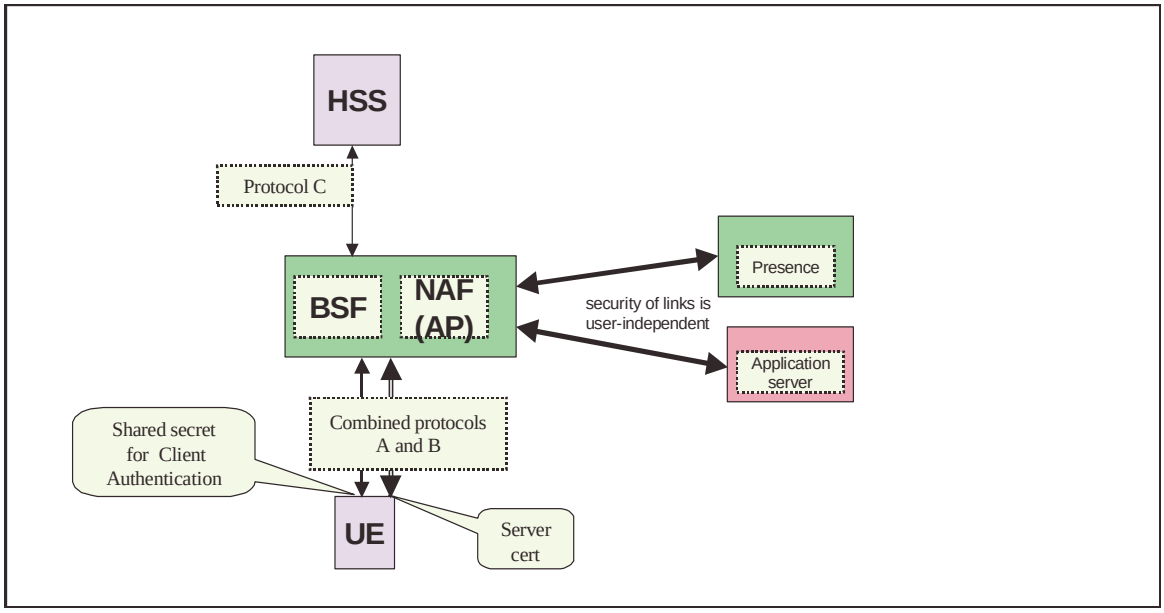

Figure 3: optimised bootstrapping of application security with co-located BSF and NAF, using https (example with NAF = authentication proxy)

Optimised sequence of events for access to co-located BSF and NAF via https

When the UE wants to access a NAF using https, and the NAF is co-located with the BSF, then the sequence of events can be as follows. Note that the NAF can also be an authentication proxy. In this case, the application servers are located behind the NAF. This subsection is based on the information flow in S3-030371, Annex A, by Ericsson.

1) The UE establishes a TLS tunnel with the NAF (if no such tunnel is in place). The NAF is authenticated to the UE by means of a server certificate.

2) If the UE does not share a key with the NAF the UE sends a http request to the NAF, containing the UE's identity (see note on identities below for alternatives). Otherwise, the UE continues with step 4).

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 may contact the (co-located) BSF to obtain a challenge and a password (shared key), computed from an AKA authentication vector. The BSF may first have to fetch authentication vectors from the HSS (protocol C). The NAF then replies to the UE by sending a 401 "unauthorized" message with a WWW-Authenticate header according to http digest akav2 (draft-torvinen-http-digest-aka-v2-01).

4) The UE now sends an http request to the NAF (or an application server behind the NAF, if the NAF is an authentication proxy), this time with an Authorization header according to http digest akav2 included. The NAF verifies the Authorization header.

5) The NAF replies to the http request returning the required information to the UE.

Note on co-location of BSF and NAF: as a matter of course, 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 3.1. 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, (as described in section 3.1), and accessed through https, when access to the NAF requires TLS (as described in section 3.2).

Note on optimisation: this optimised flow corresponds to what has been proposed for the Ut reference point in Ericsson's contribution S3-030371 for authentication proxies. It applies to any NAF accessed through https, however. Its advantage is that an extra roundtrip for client authentication using http digest is avoided. However, it appears difficult to define the general information flow of section 3.1 in such a way that its specialisation results in the optimised flow of this section. But if it is decided that this optimisation should be standardised as an option, then attempts should be made to make the option as similar as possible to the general case. This applies e.g. to the way in which UE identities are carried in http requests.

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 protocol A in S3-030341.
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.

Notes on tunnelled authentication and the use of http digest aka:

1) In this section and the previous section 3.1, different versions of http digest aka are used. This prevents man-in-the-middle attacks with tunnelled authentication. Version 1 (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-v2-01) 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.

2) 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: a) 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. b) use the specification of http digest akav2, as in draft-torvinen-http-digest-aka-v2-01, but distinguish the two use cases by different realms. All three solutions seem feasible, and the choice of these alternatives does not seem a critical point for the completion of the SA3 work on a GBA.

Note on transaction identifiers: the general approach to a GBA, as described in section 3.1, which is based on draft TS 33.109, requires the use of a transaction identifier in protocols A, B and D. The use of such a transaction identifier is neither possible nor necessary in the optimised case of section 3.2.

Proposal: The main decision SA3 has to take is whether an optimisation should be standardised as an option. The optimisation saves one roundtrip at the expense of a more complex specification and implementation. SA3 also needs to decide on the open technical issues, in particular those addressed in the notes above, i.e.

- how to carry UE identities in http requests;
- how to avoid man-in-the-middle attacks with tunnelled authentication.

Then pseudo-CRs can be written to TS 33.109.

It is proposed that the main decision be taken only when the technical issues are clear.

4. Conclusions

SA3 is asked to endorse the following proposal:

1. A GBA shall be based on TS 33.109, section 4.
2. TS 33.109, section 4, shall be modified and extended to cover the case of tunnelled authentication and of authentication proxies.
3. The material in section 3.1 of this contribution shall be taken as the basis for the modification and extension of TS 33.109, section 4. **With these enhancements, TS 33.109, section 4, shall constitute the mandatory part of the GBA specification.** Optional further additions shall not be precluded.
4. SA3 has to decide whether an optimisation, as described in section 3.2 of this contribution, shall be standardised as an option. It is proposed that this decision be taken only when the technical issues, in particular those addressed in section 3.2, are clear.
5. A GBA should respect the requirements on a GAA defined in S3z030003. An analysis needs to be written to check that the decisions taken at SA3#30 are in line with the requirements.
6. A GBA should respect the HSS-related guidelines in S3-030460. An analysis needs to be written to check that the decisions taken at SA3#30 are in line with the requirements.
7. SA3 must decide whether and how the GBA should apply to 3GPP Release 6 features.
8. Detailed security solutions for a particular 3GPP Release 6 feature shall be specified in the pertinent TS, but as much reference as possible should be made to the updated TS 33.109.
9. A GBA shall be designed in such a way that it is compatible with the use of reverse http authentication proxies.
10. A GBA shall be designed in such a way that it addresses man-in-the-middle attacks with tunnelled authentication.