3GPP TSG SA WG3 Security — S3#32 **S3-040108**
09 - 13 February 2004
Edinburgh, Scotland, UK

Agenda Item: **6.9.2 (GBA), 6.9.4 (TS 33.222 HTTPS-based services) and 6.18 (presence)**

Source: **Siemens**

Title: **Comments on S3-040043, S3-040065, S3-040068, and on Functions and Interfaces of NAF/AP**

Document for: **Discussion**

Abstract

Different Tdocs seem to have different assumptions on the functions provided by a NAF or an Authentication Proxy (AP) and about the identities sent/received on the interfaces Ua, Zn and between NAF and AS.

This comment tries to clarify different functionalities and to derive requirements on the interfaces.

1 Functions of a NAF

The functionality of a NAF may be described in a layered structure. Depending on the actual need, more levels of functionality are needed for a given NAF. The following gives four levels, where the lowest level is always required, while level 2 and 3 may be added separately. Level 4 requires either level 2 or level 3 or both.

–

1.1 Minimum functions related to GBA – level 1

Minimum function of a NAF is to authenticate a user based on the TID provided via Ua interface. The authenticity of the user is checked by sending the TID over Zn to the BSF and retrieving the necessary information (key material). Please note that the user identity, which is authenticated over the Ua interface using the key Ks_NAF, is always the private identity of the user (IMSI or IMPI), as Ks_NAF is derived from a run of UMTS AKA over the Ub interface. The TID can always be mapped onto the private identity.

No further assumptions may be made on information carried on these interfaces, in particular for Ua, as neither the kind of service nor the type of protocol used over Ua is fixed or predefined.

Properties of this “minimum” configuration of GBA are

- The only user identity the NAF possesses is the TID. No other user identity is transferred from the BSF over Zn. The NAF is assured in this case that the UE contacted and authenticated himself to the BSF before and that the BSF granted the right to NAF to authenticate this user.
- This knowledge is sufficient for admission to certain kinds of services, e.g. a “subscribers only” http based service of a MNO or a flat rate service included for all subscribers of this MNO (more specifically, to all users authenticated by this BSF).
- The TID may (additionally) act as a kind of anonymous identity or pseudonym and may be transferred from a NAF acting as authentication proxy (AP) to an AS as “asserted user identity”. This identity may even be used e.g. for charging, if some other instance (out of scope here) keeps track of TID / user identity relation for generation or translation of charging records.
- This scenario is necessary, e.g. for a NAF which is not trusted to receive a cleartext user identity.
- In particular this configuration provides the NAF with no access to any other user identity information, be it IMSI, IMPI, IMPU or any other derived identity. This is important to note, as e.g. S3-030731 and Tdocs

referring to it imply the existence of further user identity information (e.g. public ID for presence, carried in a URL), which does not exist in the general case of GBA.

Note: It is understood, that this limited knowledge of the NAF about the user identity is not sufficient for many services where the AP or the AS perform access control depending on the cleartext user identity, e.g. access to user-specific data in presence. But the basic scenario of GBA must not assume more than this minimum environment and all protocols have to work also here.

1.2 Authentication with knowledge of private user identity – level 2

For many services a cleartext user identity is needed for proper execution. This identity may be provided to NAF over Zn from BSF together with the key material and may contain IMSI, IMPI or similar identity.

This level of functionality has the following properties in addition to level 1:

- The NAF may perform access control, charging, etc. based on private identity of user (received from BSF).
- The NAF acting as AP may send this identity as “asserted identity” to AS, which may then perform the access control.

This information should be available to NAF only if privacy considerations and the policy of the MNO (operator of BSF) allows it. This decision may be based on the particular NAF requesting authentication information via Zn.

The NAF acting as AP should send this identity to AS only if privacy and MNO policy requirements allow it. Generating a pseudonym in NAF acting as AP as asserted identity and binding it to the private identity is possible. Details are ffs.

1.3 Authentication with knowledge of public user identities – level 3

If NAF wants to use public identities, pseudonyms, etc. for e.g. access control or charging these identities have to be transferred to NAF from the BSF over Zn.

This level of functionality has the following properties in addition to level 1:

- The NAF may perform access control, charging, etc. based on public identity of user.
- The NAF acting as AP may send this identity as “asserted identity” to AS.

As there are in general many public identities possible for one user, different distribution methods may be possible.

- The BSF sends a complete list of public identities (e.g. IMPUs) to the NAF.
- The NAF requests a subset of IMPUs via Zn, e.g. based on a match of service(s) requested and user profile.
- The BSF sends a subset of IMPUs e.g. based on its NAF specific policy.
- The NAF may also request public identities from HSS, if it holds any public identity of this user.This may require a Sh interface to HSS (ffs).

This information should be available to NAF only if privacy considerations and the policy of the MNO (operator of BSF) allows it. If there are different public identities bound to this user, the NAF acting as proxy needs some decision function which identity to send to the AS as asserted identity. This decision may be locally based on profiles, kind of service, etc. or may be left to BSF (and/or HSS). The sending of all public user identities to a NAF or an AS may not be allowed or desired (cf. chapter 1.4.1).

1.4 Assertion of information elements on application level – level 4

Additional functionality of NAF may be the support of application specific assertions on application level. The following example illustrates this behaviour.

Example (taken from S3-030731, requires level 1, 3, and 4): The AP inspects the URL of a PUT Request in Presence. This may be a public identity of the user and is thus not directly related to TID or private identity. The AP uses the TID to retrieve the relevant (or all) IMPUs from BSF over Zn. It then compares the public identity in the URL with the authenticated IMPU. In case of match it transfers the request to the presence AS with the asserted identity in the URL, otherwise it follows some predefined error handling routine.

This level of functionality has the following properties in addition to level 1 and level 2 or 3:

- The NAF may perform authentication for identities carried in application level information elements. This is done by binding the application layer identity to the TID related identity received from the BSF over Zn, (private or public identity/identities, cf. level 2 and/or 3).
- The NAF has to understand each particular application (or at least the syntax and semantic of the protocol to access the application). This is a kind of application specific function within NAF (cf. below).
- In particular in case of failure, the NAF has to take action as part of the application protocol. Even the removing of unasserted identities may leave a syntactically illegal request, or, even worse, a legal request with different meaning.

Because of the strong dependence on each particular application, this scenario should not be the only one considered when standardising GBA. Nevertheless GBA should support this scenario as optional. For its realisation there exist three possibilities, which are described in the following.

In the following subsections, we only consider the more general case where the AP and the AS are separate entities so that a protocol between them is required. If the AP and the AS coincide then the same functionality split remains, but no protocol is required.

1.4.1 Sending all public identities from the AP to the AS

One possibility would be that AP sends all available public user identities to AS, and the AS does the binding between one of these public identities and the identity given by the user for itself. But in general, this may contradict privacy and MNO policy requirements, in particular if AS is not tightly integrated into MNO domain. Therefore this possibility is not considered here as generally applicable.

1.4.2 Application-specific Proxy on AP

For each application requiring application level assertion there exists an “application specific plug-in” on the AP. It intercepts not only the authentication information (contained in HTTP header fields or transferred otherwise, e.g. in shared-key TLS), but also other header fields and the message body. It then binds the user specific identity (public or private) to the TID and associated private identity.

Following tasks arise here:

- In case the identity information is carried in standardised information elements and in standardized format, the application-specific proxy may be the same for all deployments of this application (with different MNOs and different domains).
- In case information element and/or format are not standardised for an application, this proxy must even take into account specific conventions of MNOs/domains.
- If no further conventions exist between NAF/AP and AS, the NAF/AP has to ensure that all possible occurrences of a user identity (of the requesting user) in a message are asserted.

Example: The presence example given above (taken from S3-030731) carries the public user identity in a non-standardized manner,. This would require a MNO/domain specific ”plug-in” in AP.

If the application specific proxy on AP handles all possible occurrences of user identities in the complete message, only then the AS may be sure that the user identity was consistently asserted by the AP. In case this cannot be guaranteed, the AP still has to add an additional asserted identity to the request to enable a cross-check by AS.

1.4.3 “Binding Proxy” on AP

To get rid of application specific functions in the AP, the protocol specific functions may be offloaded to the AS, who is (by definition) aware of the syntax and semantic of its application. This requires the AP to send a “resolve” request to the AS, in case it is configured to fulfil the binding function. A (tentative) solution is the following:

- On receipt of the message from UE the AP sends a resolve request to the AS, containing the complete user request as message body.

- The AS extracts the relevant user identity from the request and returns this identity (after necessary transformations / canonicalisations) in the response to AP.

- The AP now holds the requested identity for binding without need to know about details of the application protocol. It proceeds with the binding of the AS supplied user identity to IMSI/IMPI.

- All user requests forwarded to AS are now enhanced with the asserted public user identity, e.g. the header field “X-Forwarded-For” is added with the appropriate identity. The AP needs no other knowledge of the application protocol except to parse the HTTP header structure.

This solution allows an application agnostic AP while still allowing application specific binding of identities / pseudonyms to users.

To avoid latency, an optimized version is possible when the user supplies his complete request with the first message. Concurrent with the 401 response to the UE the AP sends the (up to now not yet authenticated) message tentatively to AS. This does not pose a security problem, as the AS does the identity extraction only, and does not trigger any further action. When the AP has completed the digest authentication, no further delay for a request to AS is incurred.

2 Requirements on Interfaces

The discussion given in the above chapter leads to some requirements on interfaces which are not yet clarified in the existing TSs 33.220, 33.222, 33.141.

2.1 Zn Interface

BSF (and/or HSS) shall have a decision function to determine which information may be sent to which NAF. This may be privacy or MNO policy driven or required by other reasons.

It may be advantageous for the NAF to have the possibility to request a subset of IMPUs only, where the selection is based on service and/or profile specific information. It may also happen that only one specific IMPU is requested.

The definition of selection criteria may also have influence on Zh interface and BSF functionality.

2.2 Sh Interface

The Sh interface is an interface between an application server and HSS. It has not been considered so far in the discussion in SA3.

It has to be investigated, in how far the use of Sh interface could be useful in this context. This is to allow the NAF/AP to retrieve e.g. public identities based on any public identity of the user received from BSF without further using BSF as relay.

It should be noted that NAF/AP must possess at least one public identity of the user to use Sh, as private identities are not allowed on this interface.

2.3 Interface between AP and AS

The interface between AP and AS has some additional requirements, for which there is no text in the current TSs 33.222 and 33.141.

- Format for transfer of asserted identity has to be defined, if this is accomplished via protocol mechanism (see ongoing discussion on cookie, or special header field).
  In case an “implicit” assertion is done (assertion of all application specific occurrences of identities in the message, cf. chapter 1.4.2), probably no specific mechanism has to be defined. It is more a contractual relation between NAF/AP and AS.

- In case a “resolve request” (cf. chapter 1.4.3) is introduced, standardization of this request may be required.