3GPP TSG SA WG3 Security **S3z030024**

3rd – 4th September, 2003

Antwerb, Belgium

Agenda Item:

Source: Ericsson

Title: Generic Authentication Architecture based on AAA and AKAv2

Document for: Discussion/Decision

1. Introduction

This document describes how Ericsson's preferred solution for protecting HTTP traffic in Presence and MBMS is related to the discussion of Generic Authentication Architecture (GAA). The generalized format of Ericsson's solution is using AAA as GAA. The document reviews concerns presented in the SA3#29 to Ericsson's solution. A new migration path is presented between AAA based GAA and Subscriber Certificates work item. Document also presents protocol details on how HTTP Digest AKAv2 can be used to implement GAA.

2. Background

Four different solutions have been proposed as a starting point for Generic Authentication Architecture (GAA) discussions in SA3: the IMS, BSF, Subscriber Certificates and AKAv2 based solutions. All of the solutions have received some support and concerns from SA3.

SA3 defined HSS-related design guidelines in order to support the design and evaluation of GAA. The leading principle of the guidelines is optimization, i.e. involvement with HSS should be minimized in terms of number of interfaces, number of AVs requested, number of requests, number of nodes using AVs, number of authentication domains and so on.

If the solution will be based on shared secrets, these requirements will quite naturally lead to the introduction of an intervening node between the application servers and HSS. Depending on the design, the node should:

1) Request AVs from HSS.
2) Cache AVs temporarily for later use.
3) Be involved with the authentication of the UE via different application servers.
4) Be involved with the generation of re-usable passwords or session keys that can be temporarily used until AKA is run next time.

These design principles have already been used in IMS where S-CSCF represents the intervening node, and different P-CSCFs represent the application servers.

The subscriber certificate based solution is a bit different because the shared secrets are only needed in the bootstrapping phase. This solution is significantly different from the rest of three, and would remove the problems related to the re-use of AKA credentials because they are only re-used for certificate bootstrapping.

Ericsson acknowledges that there are some benefits if IMS was re-used for key distribution, however, Ericsson does not consider this alternative feasible for the reasons presented already in [S3-030391] and the LS from CN1 in [S3-030325], e.g.

- Mismatch with IMS service profile
- Changes to S-CSCF in Release 6, and backwards compatibility problems
- Potential problems with key management between S-CSCF and ASs

- The load to S-CSCF especially due to variation of the services in each UE profile

BSF based solution is not seen appropriate neither because there exists more efficient and general solution to the problem, namely an AAA architecture. Subscriber Certificate bootstrapping solution could be developed in the way that BSF is replaced by AAA node, and in this way there would be no contradiction between the AKAv2 and BSF based solutions.

The rest of this document discusses the details of AKAv2 and AAA, and how they could be the basis for GAA. Migration path to subscriber certificates is also presented.

2. GAA based on AAA

The solution Ericsson has been promoting in SA3 for HTTP security (Presence, MBMS) includes the inherent idea of re-using AAA protocols [see e.g. S3-030367] as well as a generalisation of IMS AKA. The generalized version of the approach would use AAA as the intervening node/infrastructure between the HSS and application servers and/or HTTP proxy. It has a clear interdependency to WLAN access security even though WLAN and application servers may belong to different authentication domains. The use of AAA for authorization and accounting should also be further studied.

For example, if an Application Server is from a third party, the HTTP proxy should probably collect some charging information. However in the general case it is possible that the HTTP proxy does not necessarily has to produce CDRs since there is a possibility to make use of the FBC (Flexible Bearer Charging) function to produce relevant charging data if traffic can be distinguished properly e.g. through use of IP address and port number. What model to utilise is dependant on what Trust Model the Operator chooses to implement. Ericsson suggests that SA3 when progressing the work on GAA considers how GAA interrelates with the charging functionalities and other work, which is ongoing in 3GPP, which is basing the work on a Diameter infrastructure.

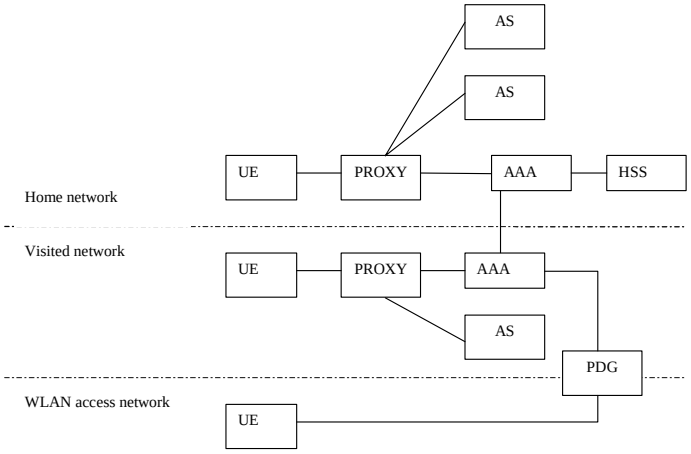

Figure 1: GAA based on AAA

Depending on the configuration, the AAA based solution could take care of the potential synchronization failure issue when the same authentication credentials are used with several applications. It is possible that the AAA server requested several AVs at a time, and cached them for later use. Different applications would be served in the order the requests arrive to AAA server. It is also possible that the AAA server just proxy the AV requests to HSS, and let the HSS handle the synchronization failure issue.

Analysis of AAA based solution using the guidelines related to HSS and GAA design is presented in Appendix A. The analysis shows that the AAA based solution is able to fulfil the guidelines quite easily.

SA3 had also some concerns for Ericsson's solution. It should be noted that SA3 has not had requirement for GAA before SA3#29, and consequently Ericsson's solution was focusing on problems related to secure HTTP access. However, as demonstrated in Figure 1, there is a logical migration path to more general authentication architecture already included.

Argument: Solution is limited to HTTP access over TLS

As explained before, the solution is based on the use of AAA protocols, and consequently is not limited to any protocol or security solution that is used between UE and application servers. The AAA function in the HTTP proxy in solution 1 can be separated as an independent entity if SA3 decides to develop GAA.

Argument: No e2e security

HTTP proxy with AAA infrastructure is able to provide e2e security if needed. For example, it is possible to redirect those requests that must not use HTTP proxy directly to the application server if the application server has access to potential new 3GPP AAA infrastructure.

It should be noted that the use of HTTP CONNECT METHOD is also possible in order to establish a secure e2e connection between the users and the AS through the HTTP proxy.

Argument: Need to develop protocols between authentication proxy and application servers

It is true that if the authentication proxy and the application servers are implemented by different vendors, the interface between them should be standardized. However, HTTP proxy that acts as a "Front Door" for application servers is fairly established concept in the industry [see e.g. Sommerland 2003]. Furthermore, there are implementations of HTTP proxy that already have the required functionality. Finally, OMA has already initiated standardization activities related to the issue [OMA-identity-management].

Argument: Replication of the proxy will create new interfaces towards HSS

This depends very much on the way the replication would be done. The smart way to do it would be to dedicate a HTTP proxy for specific UE's, and let other UE's to use a different one. Since only one HTTP proxy would request AVs for one UE, the interface towards HSS would remain the same. The other implementation alternative where one UE uses several proxies at the same time could create synchronization failures. However, this would also require parallel TLS sessions, and consequently is not appropriate to implement in practice.

3. Migration path to Subscriber Certificates

It has been argued that if SA3 continues working both with subscriber certificates and Ericsson HTTP/AKAv2 based solution, it would end-up having two incompatible authentication architectures. However, this does not need to be the case if a joint migration path for these solutions is developed.

The BSF node in Subscriber Certificates is effectively an AAA server because it authenticates the user by means of Diameter both in C and D interfaces. This is striking similarity to the AAA based solution. The only real difference is the interface A between UE and BSF. One reason for introducing interface A is the Man-in-the-middle problem related to the re-use of AKA in this context.

One new migration path from Ericsson's solution to Subscriber Certificates can be created if an AAA node replaced the BSF function, and if the interface A is removed (see Figure 2). This can be done if interface B uses HTTP Digest AKAv2. Appendix B describes the protocol details on how this can be done.

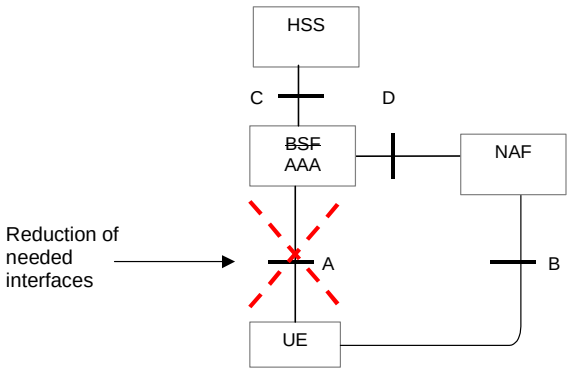

Figure 2: Subscriber certificate bootstrapping architecture with AAA

The advantages of implementing the subscriber certificate bootstrapping architecture using AAA rather than BSF are:

- AAA based solution creates a smooth migration path between the re-use of AKA credential (e.g. through the HTTP proxy) and Public Key Infrastructure.
- AAA based solution has already a solution for the roaming situations.
- AAA based solution is more efficient to use over the air-interface than BSF because interface A is not needed. This saves two round trips per AKA generated password.
- WLAN security work item could use the same AAA infrastructure.
- There is no need to develop interface A, i.e. less workload for 3GPP.
- AAA based solution fits better to IETF protocol family.

Even though earlier releases are not in the scope of this work, AAA solution has a migration path also for IMS security if needed. Basically, S-CSCF would use the Cx interface towards AAA server, instead of HSS.

4. Conclusions

This document discussed how Ericsson's preferred solution for HTTP access security could be generalized to be General Authentication Architecture. The solution is based on the use of AAA infrastructure.

The document also presents a new migration path between this solution and Subscriber Certificates work item. This migration path requires some changes to current Subscriber Certificates specifications; however, the protocol details on using HTTP Digest AKAv2 to distribute passwords and temporary identities are also presented in this document.

There are several reasons why SA3 should consider AAA as GAA, e.g.

- AAA server is able to act as a general interface towards HSS, and take care that the AKA specific synchronization failure does not occur. For example, AAA server can cache AKA AVs, and distribute them to different applications when requested.
- AAA server is able to implement the home network control by keeping the application specific passwords in the home network – but still authenticate the UE in the visited network. This feature is very valuable when designing roaming scenarios.
- AAA based solution is more efficient to use over the air-interface than BSF because interface A is not needed. This saves two round trips per AKA generated password.
- AAA is globally recognized solution for authentication.
- AAA includes also authorization and accounting. Even though these are not current requirements for GAA work in 3GPP, these features may be needed in the future.
- The AAA based solution has a migration path to Subscriber Certificates.
- AAA infrastructure is already used in 3GPP for WLAN security. The re-use of the same infrastructure is appropriate.

5. References

[OMA-identity-management] Identity Management Framework, OMA Work Item Document.

[S3-030325] LS on security solutions for the Ut reference point, source: CN1, N1-030933.

[S3-030367] Access to Application Servers using HTTP in MBMS, Ericsson, SA3, 15th – 18th July, 2003, San Francisco, CA, USA.

[S3-030391] Comparison of different approaches in the Presence/Ut interface, Nokia & Ericsson, SA3, 15th – 18th July, 2003, San Francisco, CA, USA.

[Sommerland 2003] Reverse Proxy Patterns, Peter Sommerland, 2003, EuroPLoP ‘2003, http://hillside.net/europlop/papers/WorkshopC/C6_SommerladP.pdf.

Appendix A: HSS-related design guidelines

SA3 defined HSS-related design guidelines for security architecture in SA3#29. This appendix presents an analysis on how the AAA based GAA fulfils these requirements.

1. The number of different types of interfaces to the HSS should be minimised in order to keep the complexity of the HSS low. This applies in particular to interfaces over which authentication vectors are retrieved from the HSS as they are highly security-critical.

- AAA based solution is able to minimize the number of different interfaces towards HSS to one.

2. For reasons of HSS and AuC-performance, the overall number of authentication vectors requested from the authentication centre as well as the number of requests should be kept low. Mechanisms, which make economical use of authentication vectors, should be preferred. In particular, mechanisms, which avoid bursts in authentication vector requests, should be preferred.

- The number of authentication vectors requested from the AuC is network configuration issue, and depends a lot on the architecture and services provided by the mobile operator.
- If required, AAA based solution is able to keep the number of AVs low. As a minimum, the number of requested AVs could be one.
- If required, AAA based solution is able to cache AVs and consequently minimize the number of requests to HSS.
- Re-use of HTTP Digest AKAv2 passwords would reduce the number of requests to HSS.

3. The number of nodes with access to authentication vectors should be limited in order to reduce the possibility of illegitimate access to authentication vectors.

- AAA based solution together with HTTP proxy is able to limit the number of nodes that have access to AVs.

4. The number of authentication domains (e.g. CS and the PS domain, the IMS, 3G-WLAN interworking, presence and MBMS) as well as the number of nodes within a domain for which authentication vectors for one user are stored (e.g. 3GPP AAA servers) should be kept small. This is to avoid frequent re-synchronisation. Re-synchronisation problems do not occur if unused AVs are forwarded to other nodes where they are needed, as is the case e.g. with VLRs in the same PLMN.

- The AAA based solution is able to minimize the number of authentication domains. In theory, at least presence, MBMS and even IMS could be seen as one authentication domain even without the use of HTTP proxy. AAA based solution is able to cache AVs, and consequently forward unused AVs to those nodes that need them.
- The HTTP proxy based solution is able to group the presence and MBMS into one authentication domain.

5. Mechanisms should be designed in such a way that the effect of a compromise of authentication information in one authentication domain on other domains is minimised.

- The AAA based solution is able to create service specific passwords via HTTP Digest AKAv2. Compromise of such password does not reveal passwords for other domains.

6. Authentication information should be securely stored in nodes and securely transported between nodes.

- AAA messages that carry AVs/passwords can be secured using IPsec or TLS.

Appendix B: Password and identity generation using AKAv2

This Appendix shows how HTTP Digest AKAv2 can be used to distribute HTTP Digest passwords for application servers, and to agree on end-user and server side identities. This solution can be used if Subscriber Certificate bootstrapping function is implemented using AAA instead of BSF.

HTTP Digest authentication framework use the following concepts that are also used in this section:

- 'realm': A string by which the client side knows which username and password to use. This string should contain at least the name of the host performing the authentication and might additionally indicate the collection of users who might have access. An example might be registered_users@gotham.news.com". In 3GPP/AKA context, the realm of the home network is typically stored in SIM/USIM/ISIM card.

- 'username': The user's name in the specified realm. This string is used in the server side the find the correct password for the user. In 3GPP/AKA context, the username used with the home network is typically stored in SIM/USIM/ISIM card. In most cases, the username is the same as the so-called Private Identity (IMPI). 3GPP username identifies the subscription, and for this reason the passwords are specific to the end-user device rather than to the real end-user. Note also that in normal HTTP authentication framework, the username and password are typed in by the end-user, but in 3GPP/AKA context these fields are automatically filled by end-user device.

The system has three entities: 1) the end-user device (UE) with HTTP Digest AKA implementation, 2) application server (AS), and 3) authenticator.

Bootstrapping phase

In the first step, the end-user device and the authenticator agrees on the new HTTP digest password, and optionally on new temporary end-user and application server identity.

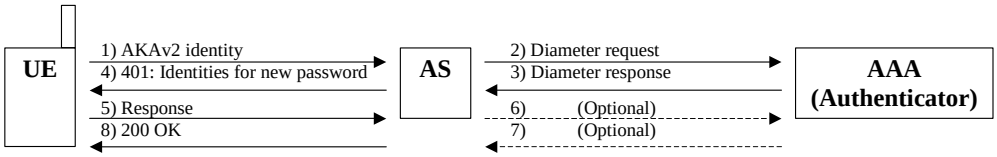

Figure 1: Password generation and identity agreement

The procedure has the following steps:

1) UE sends a request to an AS. The request is using some protocol that is able to use HTTP authentication framework.

2) Since the AS and the UE does not have shared secret, the AS requests HTTP Digest AKA authentication challenge directly from the Authenticator. The request may include the identity of the AS and the new temporal identity of the end-user.

3) The Authenticator checks if the AS is authorized to request a new HTTP Digest AKA password for this UE. If yes, then the Authenticator takes the identity of the AS and the temporal identity of the end-user (if present) from the request, and encodes this information to the HTTP Digest AKA challenge. Alternatively, the AS may also include this information to the challenge in step 4. Identity information is encoded in some standard format, e.g. username@realm. Information needed by the AS to create HTTP Digest AKA challenge are sent back to the AS. If these parameters include the HTTP Digest AKA password, then process steps 6) and 7) may not be needed.

4) The UE is challenged by HTTP Digest AKA authentication challenge. The AS may add the identity information (both the AS and end-user identities) to the authentication challenge before sending the challenge to the UE. However, in this case, the AS must be the end-point for the authentication, and possess the HTTP Digest AKA password.

5) The UE authenticates the network (as defined in standard AKA protocol) and generates a new password based on HTTP Digest AKA challenge. The UE stores locally the identity of the AS (e.g. the "realm"), the newly generated

password, and the new temporary ‘username’ for itself to the AS to be used later for mutual authentication with the AS – if present in the challenge. UE sends the authentication response to the AS.

Note: The UE should mark the potential new ‘username’ as temporary username. This username and related HTTP Digest AKA password should be removed when new ‘username’ and password is generated for the same realm. If the challenge does not include new temporary username, then the existing username should be re-used.

6) If the AS did not receive the end-user password in step 3), it must request Authenticator to do the authentication at this phase.

7) If the AS did not receive the end-user password in step 3) and it requested authentication in step 6, the Authenticator authenticates the UE, and returns the appropriate result to the AS. The Authenticator may also send the end-user password to AS at this or some later phase.

8) If the UE authentication was successful, the service is delivered to the UE.

Use of the password

In the second step, the end-user device and the application server can start using the new HTTP digest password, temporary end-user and application server identity.

When the AS wants next time to authenticate the UE (which may be directly after the previous procedures, or after some longer period of time, e.g. when the UE contacts the AS next time), the procedure is as presented in Figure 2.

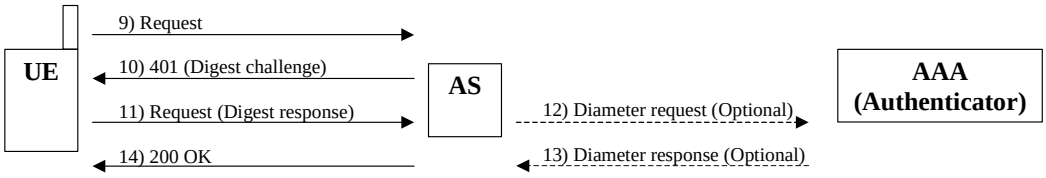

Figure 2: The use of new password.

9) UE sends a HTTP request (typically HTTP GET) to an AS.

10) Since the AS and the UE have now a shared secret, the AS challenges the UE with HTTP Digest challenge. The challenge includes the identity of the AS in the “realm” parameter.

11) The UE sends an authentication response (typically in HTTP GET request) back to the AS using the new temporary ‘username’ and password for the AS created during the previous phase. If the new temporary username was not created, then the normal AKA specific username is used. The UE uses the “realm” parameter to identify the correct password.

12) If the AS possesses the end-user password, steps 12 and 13 are not needed. If not, then the AS request from the Authenticator (or some other network entity to where the Authenticator has stored the UE specific password), for authentication.

a) 13) The Authenticator returns information on whether the authentication was successful or not.

b)

14) If the authentication was successful, the AS delivers the service to the UE.