3GPP TSG SA WG3 Security — S3#28 **S3-030256**
6 - 9 May 2003
Berlin, Germany

Title: **Analysis of HTTP authentication**

Source: **Nokia**

Agenda item: **Presence**

Document for: **DISCUSSION AND DECISION**

This paper analyses 3 approaches proposed to last S3 meeting, and concludes a way to forward.

1. THE ISSUES IDENTIFIED IN S3-030056

S3-030056 mandates the registration prior to HTTP download. In S2 LS (**S2-031583)** states: "Usage of Mt does not require that the user is IMS registered (though in most cases the UE will be registered)." This suggests it is not a correct assumption to mandatory IMS registration. From network architecture point of view, the UE bypasses IMS and contacts Application server directly. Thus the data manipulation operations should not be tight to IMS registration.

Another reason is that, for R6 and further releases, when non-IMS UA is supported, there may be the needs for non-IMS UA to access Application Server (AS) but never registering to IMS. One example is the Laptop PC as the UA to manage the user data via HTTP connection.

If the UE is de-registered, the HTTP authentication would be based on last registration secrets. Suppose if the UE is de-registered, the HTTP authentication is based on last registration secrets, the UE must derive the key even though it may never used. This is because IMS does not re-use old keys when initially registers.

If the mobile is power off and on again, then it has no valid CK stored in memory or ISIM and can not access HTTP unless it registers IMS. This is another concern towards the IMS registration dependency.

Conclusion: In general, based on these reasons, we consider it is not reasonable presumption that UE is registered.

Next section we'll look at the proposal from 2 aspects, one is forwarding BSF generated credentials to Applications Servers, and the other is to utilize the credential for transport security for Mt interface (UE- Applications Servers).

1.1 Forwarding BSF generated credentials to Applications Servers

1. The support from current network architecture:

   First, ISC interface (between AS and the S-CSCF) does not support for security key distribution, and service profile delivery. To date Stage2 (SA2) or stage3 (CN1) do not have support for this. In addition, it would most probable require IETF work (new event package SUBCRIBE/NOTIFY mechanism), thus introduce dependence to the outside IETF progress.

   Secondly, the solution introduces confliction with R5 architecture, where CK (cipher key) is delivered to P-CSCF from S-CSCF. Siemens paper prohibits sending to P-CSCF the CK, which is now replaced by a derived key from CK. This approach seems to introduce the backward incompatible with R5 IMS implementation.

2. The connection between IMS and Application servers:

   S2 LS states a principle for designing, that "The possibility to have multiple Application Servers serving the same user at the same time should be taken into account."

   Basically the S3-030056 proposes multiple-to-one connections between each unique AS and the S-CSCF. This is redundant design compared to Nokia Authentication Proxy approach (S3-020528) as well as Authenticator in Ericsson's proposal S3-030084 where a unique connection between AP and S-CSCF is re-used by all application servers. And only once the distribution of profile information is needed.

3. The combination of 3G specific and SIP servers:

   S1 stage 1 specification TS 22.228 on Presence states in chapter 6: "IP multimedia applications shall, as a principle, not be standardised, allowing operator specific variations. It shall be possible to enable rapid service creation and deployment using service capabilities. It is important that commercially available IP multimedia applications are supported. In general compatibility shall be with these IP multimedia applications instead of building 3GPP-specific solutions." We think that Applications Servers mandatory 3GPP authentication secret imposes additional restriction to the requirement in 22.228. Mt interface protocol is HTTP, it should not be assumed that SIP Application Server understanding the HTTP. An authentication proxy would help much better the case.

In Nokia proposal, the AP is a part of operator's network; it is fully trusted by HSS and S-CSCF. Applications Servers do not know anything about ciphering key for transport security, but Applications Servers s knows the UE's public identities. The private ID can be revealed to Applications Servers if required. Introducing AP as a part of security functionality interfaces with existing SIP application servers and IMS components would fulfil this requirement.

4. The key for the link between UE and the Applications Servers

S3-030056 proposal may have 2 alternatives:

a. All Applications Servers share the common key. The S-CSCF MUST push the secret based on last successful registration to EVERY AS, each time after the authentication. Otherwise the key used is synchronized for each AS. This introduces a lot of unecessaory traffic between the S-CSCF and Applications Servers that is wasting of resources.

b. Each AS gets a distinct key derived from the same CK. We foresee a security disadvantage in future, when new Applications Servers are introduced, the derivation function may not be able to produce keys completely orthogonal. Furthermore, there are no designing neither cryptographic analysis available for this. Finally, this also suggests changes to S-CSCF once new application is added.

Conclusion: Based on these reasons, we consider current architecture does not support re-use of the bootstrapping through IMS registration. And the connection between IMS and the application servers could be optimised by introducing a proxy that is trusted by the IMS.

1.2 Utilization of the credential for transport security for Mt interface

How to use the IMS key derived key, their may be 3 possibilities:

- For WTLS, using the shared-secret handshake. SHARED_SECRET key exchange suite is requested by the client in ClientHello message.

But we should not propose to use WTLS for R6 service. After all it's an old protocol, and WAP 2.0 in fact utilizes the TLS that is current working assumption taken by last S3 meeting .

- As TLS/SSL transport key, it is not possible to use pre-shared secret as master secret without a full handshake procedure.

- As password in HTTP Digest authentication. This would still require transport protection.

Conclusion: The TLS must be established, and credential has to be used in HTTP level inside the tunnel.

2. COMMENTS ON S3-030084

Ericsson proposl S3-030084 presents a network element called Authenticator, that is resemble to Nokia's AP. They suggest to authenticating independently the UE again for HTTP connection, this is close to Nokia's proposal in general.

3 alternatives are proposed for HTTP authentication:

- the HSS forwards a standard Quintet towards the Authenticator. One new function f1(K, A-Data)=A-Data-MAC is required to distinguish from regular AKA for IMS and UTRAN authentication. This requires R5 HSS to add one more dimension, i.e. A-Data-MAC as new random in the challenge. Note A-Data can not be shared among the User group.

Comments:

- HSS handling A-Data will request the change to Cx interface, as well as implementation in HSS and S-CSCF. If S-CSCF generates and handles the A-Data, the HSS will not have a copy of A-Data. This means if terminal is de-registered, the network may not have a copy of keys anywhere.

- No matter what functionality is, it should not re-use the AKA f1 used as MAC generation function in UICC.

- Delivery of A-data to the UE requires change to R5 IMS accessing, say, HTTP Digest AKA.

- To populate the A-Data-MAC again requests change to Digest AKA.

- USIM or ISIM application in the UICC would need to be changed.

- the requirements of tunneling protection and server authentication should be mentioned, otherwise just updating AKA to AKAv2 is not enough to prevent untunnelled modes.

- If the K is modified to be IK or CK, and let the Authenticator/Authentication Proxy to handle the A-Data, many of the problems would go away. See section 4.1 further analysis.

    - a service identifier SId to generate derived XRESd, CKd and IKd with the help of a pseudo random function: (XRESd||CKd||IKd) = PRF(SId|| XRES||CK||IK).

Comments:

- This solution is similar with Siemens proposal, a similar optimization problem described in 4b in section 2.1.

- The authenticator can fetch a regular AKA quintet from HSS and using PRF to generate session parameters. PRF should not be in HSS since this would require dramatic change in Cx interface.

- If Sid = constant, i.e. public value, it's not need to deliver to the UE. The using Digest response with IK or Digest AKA response with XRESd to authenticate the UE (over TLS), this approach is neat comprared to the first way. Because it does not require changes to HTTP Digest or IMS existing entities. This actually derives password from bootstrapped authentication, similar to way d in section 4.1 analysis.

- But the CKd and IKd seem to be not useful, since usually TLS does not allow the transport session key be pre-shared.

3. DISCUSSION

First of all, the certificate proposed in S3-030060 is the most secure way to authenticate UE. Though it may introduce schedule dependency, it has no flaw technically. In the long run it brings benefit as IETF compliant solution and a platform to enable all other services such as presence.

Another approach is to rely UE authentication to the BSF function. Since the functionality would re-use the existing protocol such as Digest AKA, there should not be problem for the function completion in R6.

3.1 MitM attack mitigation

Regarding how to avoid the MitM attack: basically the AKA authentication must be bind to the context. This can be done in several ways:

Assume K = bootstrapped session key (IK|CK), and A-Data-MAC is generated from MAC (K, A-Data)

a) binding AKA authentication and TLS master key
- A-Data is the TLS master key
- TLS master key is either selected by the server, or jointly by server and client, but not by client alone)

b) binding AKA authentication and TLS session ID
- A-Data is the TLS session ID
- TLS master key is selected by the server or jointly
- This requires client to perform TLS server authentication

c) binding AKA authentication and TLS server's DNS name
- A-Data is the TLS server's DNS name
- This requires client to perform TLS server authentication

d) binding AKA authentication to the specific tunneled protocol
- A-Data is such that A-Data-MAC is not used outside this protocol. This would go to a common value.
- This requires client to perform TLS server authentication

For all of these cases,
a) A-Data-MAC can be sent as a separate information element in the client authentication protocol.
b) A-Data-MAC can be used as the "password" in HTTP Digest

In TLS Server authentication + HTTP Digest with bootstrapped key as password approach, this doesn't need any separate binding if we can include some PMG specific string in every HTTP message related to PMG. In other words, this is an instance of scenario "d" above.

4. PROPOSAL

In brief, we propose to select the TLS approach with server-only authentication, and then using HTTP Digest with bootstrapped key or HTTP AKA with password derived from bootstrapped key for client authentication.

The Authentication Proxy can either retrieve from BSF the bootstrapped key or from HSS directly, preferably from HSS.