3GPP TSG SA WG3 Security — S3#22 **S3-020135**

25 - 28 February 2002

Bristol, UK

Source: **Qualcomm**

Title: **Problem with use of RES in Digest-AKA**

Document for: **Discussion**

Agenda Item: **IMS (7.3)**

TDOC S3-020089 introduces draft-niemi-sipping-digest-aka-00.txt, describes a mechanism whereby AKA is used to create a shared secret to be used as the "password" for HTTP-Digest authentication for SIP. The draft uses "RES", derived from the AKA process. TDOC S3-020067 then introduces draft-undery-sip-auth-00.txt, which builds upon the previous draft, for authentication of SIP signalling messages.

It is expected that RES, as derived during AKA computations, will generally be shorter than the 128-bit key from which it is derived. Typically, RES might be as small as 32 bits. (Note that IMS is expected to be backward compatible with existing USIMs.)

The use of RES, with its reduced entropy, as the "password" for HTTP-Digest introduces a "choke point" in the computation of the various digests. This document first describes an attack based on this, then proposes a modification to draft-niemi-sipping-digest-aka-00.txt which would appear to address the problem.

Choke Point attack

In a normal flow of events, the following steps accomplish the authentication of SIP messages:

1. UE attempts to register
2. Attempt is rejected because it is unauthenticated; rejection message carries AKA-related information and an HTTP-Digest nonce
3. UE/USIM checks the AKA-related information, and uses it to compute RES.
4. RES becomes the "password" shared between the UE and CSCF.
5. UE computes the response to the HTTP-Digest based on RES, and attempts to register again.
6. This time it is successful.
7. … time passes …
8. UE wants to send another SIP message (eg. Invite) and the HTTP-Digest method calculates authentication information based again on RES (actually based on A1, which is derived from RES in step 5 above, but that's a detail).

Now imagine an attacker who wishes to break the integrity protection of a subsequent SIP message, having intercepted the traffic to date. The attacker does:

1. intercepts the messages in steps 2 and 5 above.
2. All of the information used in the calculation of the response in step 5, except for the value of RES used, is present in these messages. The attacker tries out the (hypothetically) 2^{32} values of RES, attempting to duplicate the response. With very high probability, he will succeed with exactly one candidate value for RES, in the time needed to calculate 2^{31} MD5 hashes (about 5 minutes on my laptop).

3. Using this value of RES, the attacker can now forge SIP messages, or alter messages in transit, recalculating the Digest after altering the message.

Conclusion and solution

The problem here is that RES has less entropy than the actual shared secret, and allows attacks. Instead, one of the values derived from AKA that has full entropy should be used. We recommend use of IK for this function, instead of RES, since it seems to be the natural candidate.