3GPP TSG SA WG3 #22bis aSIP ad hoc **Tdoc S3z020055**
Fort Lauderdale, Florida USA
8th April 2002

Source: **Nokia**

Title: **Use of MD5 with IMS AKA**

Agenda item: **7.3**

Document for: **Discussion**

The approach of using HTTP Digest framework to utilize 3GPP AKA in IMS has created an extensive discussion on the SA3 mailing list. The IETF53 meeting in March has expressed preference on the approach in draft-niemi-sipping-digest-aka-00 against an alternative approach of trying to use HTTP Authentication solely as a transport protocol for AKA messages. This would mean that effectively the cryptographic computations in AKA are slightly modified.

As expressed by Siemens in the email discussions: “`If RES is used in draft-niemi-... only once, i.e. only in the direction from the UE to the S-CSCF, and is not used in the opposite direction to provide server authentication nor used in a second run of the http digest protocol`
`then the problem with the proposed use of RES boils down to the fact that the function f2 which is used in the AKA protocol to derive RES from the long-term key K and RAND is replaced with f2 concatenated with MD5, with some additional input (RAND|AUTN=nonce, client-selected cnonce). As I said in my first email, I see no immediate attacks for this restricted use of RES, but I have a more general worry about changing the authentication functions.`”

The purpose of this document is to give further evidence that no attacks can be created because of this modification.

Firstly, we note that the information in the messages of draft-niemi-…, i.e. MD5 hash values do not give any advantage on trying to find any of the secret parameters in AKA. This is simply because in the case of “plain 3GPP AKA” the RES becomes public and any attacker who sees the RES can compute the same information , i.e. the MD5 values of draft-niemi-.. also by himself.

Secondly, as regards to network authentication executed by the UE there is no difference between “plain” AKA and Digest AKA, since the authentication can be done even before RES is created in the UE side. In other words, the whole feature is based on verification of RAND and AUTN and these are sent in plaintext in both cases.

Thirdly, as regards to user authentication by the network, the basic question is the following: “Is the probability, for an attacker who does not know the master key K, to get a response message nevertheless accepted by the network greater in Digest AKA case than in the “plain” AKA case? “
In both cases the attacker can always guess the RES and prepare the response message accordingly. If the length of RES is n then the probability of success is 1/(2^n).
The network server accepts the authentication in the Digest AKA case only if the MD5 hash value in the response message is the same than the corresponding hash value calculated in the server using XRES as a password.
In principle, it can happen that there are two XRES values that yield the same MD5 hash value. In this case, the attacker could get a response message accepted with a slightly better probability than guessing the RES directly. The problem however is that the attacker must first find out which MD5 value is implied by two different XRES-values. In other words, the attacker has to find a MD5 collision. Furthermore, the collision has to be of a specific type: the two input messages that create the collision (a) differ only in the password part and (b) contain the specific nonce created by the network.
Now, (a) implies that the search for collision is much harder while (b) implies that the search can begin only after the nonce has been seen by the attacker, i.e. the search must be performed while on-line !

The MD5 algorithm is not recommended for general use as a one-way function, since there is a widely-accepted suspicion that collisions may be found. However, it should be stressed that NO actual MD5

collisions with ANY two inputs are yet reported in the literature. A tremendous break-through in MD5 cryptanalysis has to happen before the attack on Digest AKA described above is feasible.
Even in the (highly unlikely) case where special MD5 collisions can indeed be found on-line, the probability of a successful attack is only doubled. It is clear that it is far easier to double the attack success probability by simply trying twice !

To improve the probability of successful attack more dramatically, a joint MD5 collision of many input messages must be found. Even assuming a tremendous break-through in MD5 cryptanalysis this does not seem to be possible.