3GPP TSG SA WG3 Security#32 **S3-030044**
9 – 13 February, 2004
Edinburgh, UK

Agenda Item: **6.9.2 (GBA)**

Source: **Siemens**

Title: **Informational annex on the use of parameter n – Pseudo-CR**

Document for: **Discussion and decision**

Abstract

In the current version of the Generic Bootstrapping Architecture specification (TS 33.220 v100), a parameter n is introduced to define an identity NAF_Id_n, which is input to the derivation of key Ks_NAF. It is proposed in this contribution to include an informative annex in TS 33.220 to explain the usage of n as it is believed that this may help users of the specification to correctly configure their systems.

TS 33.220 v100, section 4.3.2, contains the following text:

“
Ks_NAF is computed as Ks_NAF = KDF (Ks, key derivation parameters), where KDF is a suitable key derivation function, and the key derivation parameters include the user's IMSI, the NAF_Id_n and RAND. The NAF_Id_n consists of the n rightmost domain labels in the DNS name of the NAF, separated by dots (n= 1, ..., 7). For n = 0, NAF_Id_n equals the full DNS name of the NAF. The next bullet specifies how the UE obtains n.

NOTE: This note gives an example how to obtain the NAF_Id_n: if the DNS name of the NAF is "server1.presence.bootstrap.operator.com", and n = 3, then NAF_Id_n = " bootstrap.operator.com".
“

It is proposed to include the following text as a new Annex into TS 33.220:

Annex C (informative): The use of key derivation and the parameter n

Section 4.3.2 of this specification states that a key Ks_NAF is derived from the bootstrapped key Ks. It further states that the key derivation function takes a parameter NAF_Id_n as input. NAF_Id_n is computed from the DNS name of the NAF and a parameter n as specified in section 4.3.2, i.e. NAF_Id_n consists of the n rightmost domain labels in the DNS name of the NAF, separated by dots. This Annex explains the rationale for key derivation, and gives guidance as to how determine an appropriate value of the parameter n.

The purpose of the optional key derivation specified in section 4.3.2 is to give the possibility to separate the space of NAFs into security domains. Please note that key derivation alone would not necessitate the introduction of the parameter n, as one could set NAF_Id_n always equal to the full DNS name of the NAF. This is achieved according to this specification by setting n = 0. The separation of security domains is further explained in section C.1 below.

The purpose of the introduction of the parameter "n" is the possibility to obtain a single key Ks_NAF (and hence e.g. a single shared-key-TLS tunnel or a single http digest security session) between a UE and a NAF, even when multiple DNS names map to the same NAF.

C.1 Separation of security domains: it is the intention of key derivation to thwart an attack, where a compromised NAF1 with low security requirements impersonates a NAF2 with high security requirements towards the UE, using the compromised key Ks. If key derivation is used, and if values NAF1_Id_n and NAF2_Id_n are different, then Ks_NAF1

and Ks_NAF2 are also different, and a compromised key Ks_NAF1 cannot be used to impersonate NAF2. In other words, when key derivation is used the following requirement can be satisfied:

(REQ1) *It shall be possible that the set of NAFs is separated into security domains that a key Ks_NAF valid in one security domain cannot be used in another security domain.*

Example: assume that n = 3, i = 1, 2, ... and that there are servers with names
serveri.gaming.operator1.com;
serveri.ssc.operator1.com.
Then REQ1 is fulfilled for the two security domains, "gaming" and "ssc".
Note that REQ1 would also be fulfilled if n>3.

C.2 One shared key for one (UE, NAF) pair: REQ1 is satisfied in particular if always the full DNS name is used as input to key derivation. This results in different keys Ks_NAF for different DNS names of a NAF. Such a behaviour may not always be desirable. E.g. when using authentication proxies (APs) the AP may take the role of a NAF. It may happen (depending on the network configuration) that two application servers (ASs) with different DNS names sit behind the same AP, and it may be desirable then that the UE is able to recognise this fact and use the same shared key with the same NAF, cf. also TS 33.220 v100, section 4.3.3, 1st bullet. If not there could be as many different keys shared between UE and NAF as there are DNS names mapping to the same NAF. The introduction of the parameter "n" gives the possibility to configure the system in such a way that the following requirement REQ2 is satisfied:

(REQ2) *It shall be possible to choose DNS names and the value n in such a way that DNS names mapping to the same NAF have the same NAF_Id_n.*

As only NAF_Id_n and not the full DNS name of the NAF is input to the key derivation procedure only one key Ks_NAF is produced for one NAF at a time if REQ2 is satisfied. When the UE starts communicating with a NAF it first checks whether it already has a key Ks_NAF for the corresponding NAF_Id_n. If yes, the UE immediately starts secure communication with the NAF using this key and supplying the corresponding transaction identifier (used as the user id in http digest).

Example: let n = 4, assume that there are servers in the network of operator1 with the following DNS names:
server1.proxy1.presence.operator1.com;
server2.proxy1.presence.operator1.com;
server1.proxy2.presence.operator1.com.
Then, clearly, REQ2 is fulfilled. But note that REQ2 is also fulfilled when n < 4.

C.3 Can the same NAF_Id_n be used for different NAFs?

When n = 3 in the example in C.2, then all the servers behind proxy1 OR proxy2 have the same NAF_Id_n = "presence.operator1.com". Assume that the UE first accesses server1.proxy1. presence.operator1.com and next server1.proxy2. presence.operator1.com . The UE believes that it already shares a key with server1.proxy2. presence.operator1.com , which is not the case. Now, this need not be a problem , as the NAF could simply proceed to fetch the missing key, using the supplied transaction identifier. Only if the BSF has already deleted the corresponding Ks, a new run of the bootstrapping procedure will be triggered. But the resulting overall behaviour may be not optimally efficient as shown by the example of http digest:

Example: when the UE switches from one NAF to another with the same Ks_NAF then the new NAF will not accept an Authorization header sent by the UE in the first message to the new NAF because the new NAF does not recognise the nonce as fresh. Hence the new NAF will reply with the flage "stale" set to TRUE in the www-authenticate header, cf. [rfc2617, section 3.2.1]. This would add another roundtrip, but not cause a protocol failure. How serious a disadvantage this is depends on the frequency of switching between different NAFs with the same NAF_Id_n.

If the described behaviour is considered undesirable it can be avoided by configuring the system in such a way that additionally the following requirement is satisfied:

(REQ3) *DNS names and the value n shall be chosen such that DNS names mapping to different NAFs have different NAF_Id_n.*

C.4 The choice of 'n' and its impact on the DNS name space

REQ1 addresses the security concerns mentioned in C.1 above. REQ1 alone does not lead to any restriction on the assignment of DNS names.
The restrictions imposed by REQ1 and REQ2 combined still seem to leave room for a lot of flexibility in the

assignment of DNS names.
When, in addition, requirement REQ3 is to be satisfied restrictions on the assignment of DNS names are most severe.

Please note that the BSF determines n, but the GBA does not know for which NAF the UE wants to use the bootstrapped key Ks. So n cannot be chosen specifically for particular NAF. It therefore has to be the same for all NAFs, and, in general, n would be the same in all protocol runs involving one BSF. But different BSFs accessed by one UE may use different values n, and one BSF may decide to change the value n, e.g. when the policy or the network configuration changes. Please also note that the NAF does not know n, as the NAF obtains the derived key from the BSF.