3GPP TSG SA WG3 Security **S3-030374**

15 – 18 July 2003

San Francisco, USA

Agenda Item: 7.1

Source: Ericsson

Title: Key Expansion function for IMS/Presence

Document for: Discussion/Decision

1. Scope

This document presents two possibilities for key expansion function for IMS/Presence. An accompanying CR to this discussion paper is attached and Ericsson proposes that SA3 endorses the CR.

2. Background

According to [1] Sect 8.1.1, the encryption key CK_{ESP} shall be obtained from the key CK_{IM} established as a result of the AKA procedure, using a suitable key expansion function, the latter to be specified in Annex I. Below, we study two alternatives for key expansion functions meeting the above length requirements and their security properties.

We first note that the key CK_{IM} is 128 bits in length, and that the specified DES-EDE3-CBC algorithm requires 192 bits of key material, whereas the later-to-be-added AES algorithm requires 128 bit keys.

We make the following basic assumptions:

- The option of modifying the use of AKA to derive more key material is ruled out.
- Any future need for keys longer than 192 bits is unlikely.
- Since CK_{IM} is 128 bits, an effective key size of 128 bits (112 for 3DES) is both necessary and sufficient for CK_{ESP}.

3 Key expansion

3.1 Key expansion function 1

Divide CK_{IM} into two blocks of 64 bits each:

$$CK_{IM} = CK_{IM1} \| CK_{IM2}.$$

The key for DES-EDE3-CBC is then defined to be

$$CK_{ESP} = CK_{IM1} \| CK_{IM2} \| CK_{IM1},$$

after adjusting parity bits to comply with [3]. This is known as two-key triple-DES, and is fairly standard.

Properties: Two-key triple modes of encryption should always be used with caution as one often do not get the "full" security one hopes for. The threat lies in various forms of dictionary attacks and time-memory tradeoffs. For instance, as is well known (see e.g. Fact 7.40 of [4]), two-key 3DES in ECB mode does not give 112-bit security under time-

memory attacks using chosen plaintexts. Specifically, using t chosen plaintexts, an attack on two-key 3DES-ECB is possible that uses on the order t memory and $2^{120 - \log t}$ operations. For instance, would it be possible for an attacker to choose 2^{40} plaintexts, the attack complexity is about 2^{80}. For CBC, which is the case here, it is noted in [7] that two-key 3DES-CBC is not more secure than (single) DES-CBC under known plaintext attacks using *all* 2^{64} plaintexts. On the other hand, not even three-key 3DES-CBC resists such dictionary attacks. We do not believe that these attacks are "real" serious threats.

Therefore, the main drawback of the construction is that it does not generalize in a natural way to other key-sizes (though as noted, we do forsee a demand for larger keys). An advantage of the scheme is of course its simplicity.

3.2 Key expansion function 2

This is a more elaborate scheme. We first note that according to [2], we can assume that HMAC-SHA1 is already in place in Release 5 for the purpose of integrity protection. The proposal reuses this function as a pseudo-random function (PRF). The proposal is essentially identical to [5], which in turn is a slight modification (simplification) of the PRF used in TLS, [6]. Conceptually, the construction can be thought of as running HMAC-SHA1 in "Output Feedback Mode", and in addition, masking each output by an extra application of HMAC-SHA1. For self-containment, we reproduce the specification below.

We first specification the so-called Ph-function. This is a component of the construction, and we define

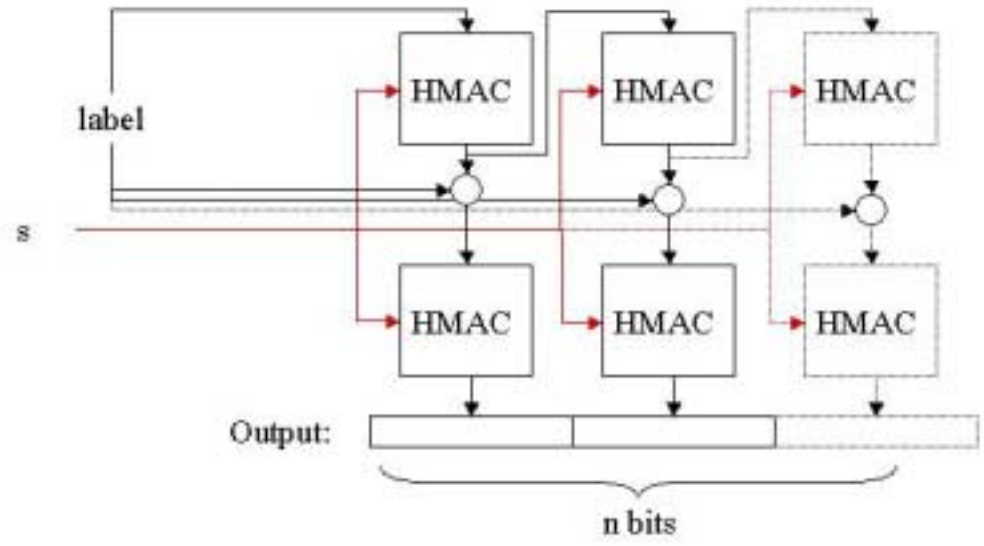
$$Ph(s, label, m) = HMAC(s, A_1 \| label) \| HMAC(s, A_2 \| label) \| ... \| HMAC(s, A_m \| label)$$

where it is to be understood that HMAC is based on h being the hash function SHA1, and label is some string and

A_0 = label,

A_i = HMAC (s, A_{i-1}),

see also the figure below. Note that the output of Ph is n = 160m bits long. Thus, m = 1 or 2 will suffice for most practical purposes we foresee.



Now, given Ph, the PRF is defined as follows. Let in_key be the input key, which is k bits in size, and let n be the desired length of the output key. (In the application in mind, k = 128, n = 196.)

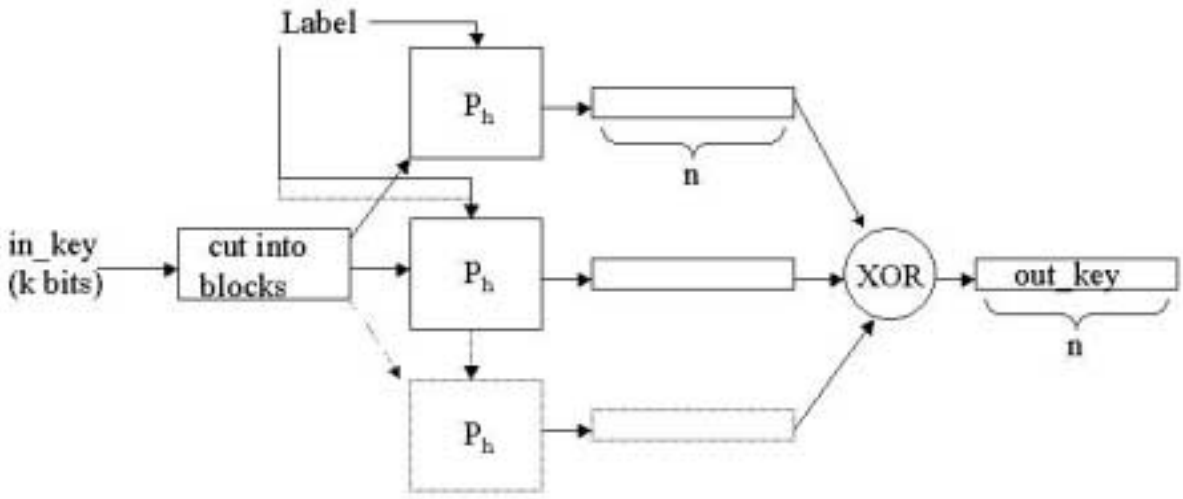
1. let b = k / 512, rounded up to the nearest integer (for the above parameters, b = 1)
2. split the in_key into b blocks, inkey = $s_1 \| ... \| s_b$, where all s_i, except possibly s_b, are 512 bits each

3. m = n / 160, rounded up to the nearest integer (for us, m = 2)

Then, the output key, out_key, is finally obtained as the n most significant bits of

$$PRF(in\_key, label, m) = Ph(s_1, label, m) \text{ XOR } Ph(s_2, label, m) \text{ XOR } ... \text{ XOR } Ph(s_b, label, m),$$

see figure below.



The value of "label" is rather arbitrary but could e.g. be the string "IMS_ESP_ENCR_KEY" as an ASCII string, etc. If other keys are to be derived from the same in_key, this is done by using distinct labels for each derived key.

Properties: The difference to the TLS version is that that uses two hash functions (SHA1 and MD5), runs to copies of Ph based on these hashes and half of the in_key each, and then XORs the two results. As SHA1 is well-studied, we see no problem in relying on SHA1 only, which considerably simplifies the construction. As far as we know, no weakness is known in the above construction. Of course, when the derived key is later used for 3DES encryption, generic attacks on triple modes of operation (e.g. dictionary attacks as mentioned above) are still possible.

The construction might seem complex, but on the other hand if offers great flexibility in supporting various input/output key sizes. Also, the complexity in practice is not so high since, as noted, b = 1, m = 2 suffices for our purposes.

3.3 Other alternatives

There very are few well-studied, standardized key expansion functions. One could consider designing a new function for our purposes. However, we feel more comfortable relying on something that has received some amount of public scrutiny. An alternative might also be to use e.g. AES in counter-mode, but as AES is not yet part of the TS, it is simpler to reuse something already in place from Release 5. Also, when AES indeed is added, use of DES is likely to decrease, and AES will be used directly also for the confidentiality (without need for any key expansion, see below).

A possibility worth mentioning, however, is the following from [8]. It is quite simple but also quite tailored to 3DES. Let $CK_{IM} = CK_{IM1} \| CK_{IM2}$ be as above and let X_1, X_2 , X_3 be three 64-bit constants. The three keys K_1, K_2 , K_3 are derived as:

$$K_j = E(CK_{IM1}, D(CK_{IM2}, E(CK_{IM1}, X_j)))$$

for j = 1, 2, 3 and where E/D denoted DES encryption/decryption. That is, two-key 3DES is used to derive the keys. Note that the security is still bounded by the 128-bit size of CK_{IM} and the dictionary attacks mentioned above apply according to [7].

3.4 Key material for AES

We propose that when AES is later added to the specification, CK_{IM} is used directly as key for the block cipher. The security then depends only on the security of the AKA algorithms. It should be noted that although the AKA-Milenage set of algorithms are also based on AES, we do not see any problem in reusing AES in this way. If for security reasons, SA3 foresees a future deployment of AES-192 or AES-256, it still hard to see the need for a key expansion function for AES, since there would be no increased security unless the AKA algorithms are able to produce the corresponding key material directly.

3 Conclusions

Unless attacks using on the order 2^{64} memory/known plaintext are considered an issue, we recommend SA3 to consider using two-key triple DES and its very simple key expansion function as above and to as soon as possible promote the use of AES directly with CK_{IM}. Ericsson proposes that the attached CR is accepted by SA3.

However, if SA3 believes that it is likely that other keys in the future needs to be derived from CK_{IM}, Ericsson recommend the second alternative and that this is then further progressed in future SA3 meetings. Then the corresponding sections from above should be included in the Presence TR as a placeholder since no corresponding CR is presented at this meeting.

4. References

[1] 3GPP TS 33.cde: ”Presence Service; Security (Release 6)”.

[2] 3GPP TS 33.203: “3G security; Access security for IP-based services (Release 5)”.

[3] RFC 2451 (1998): "The ESP CBC-Mode Cipher Algorithms".

[4] A. Menezes, P. van Oorschot, and S. Vanstone: “Handbook of applied cryptography”, CRC Press 1997.

[5] J. Arkko et al.: “MIKEY: Multimedia Internet KEYing”, draft-ietf-msec-mikey-06.txt, work in progress.

[6] RFC 2246 (1999): “The TLS Protocol”.

[7] E. Biham: “Cryptanalysis of Triple Modes of Operation”, Journal of Cryptology **12**(1999):3, 161-184.

[8] B. Schneier: “Applied Cryptography”.