3GPP TSG SA 3 Meeting #10
Antwerp, Belgium, 19-21 January 2000

***Document* S3-000084**
e.g. for 3GPP use the format TP-99xxx
or for SMG, use the format P-99-xxx

DRAFT CHANGE REQUEST

Please see embedded help file at the bottom of this page for instructions on how to fill in this form correctly.

33.105	CR	006	Current Version:	3.2.0

GSM (AA.BB) or 3G (AA.BBB) specification number ↑ *↑ CR number as allocated by MCC support team*

For submission to:	TSG SA #7	for approval	X	strategic		(for SMG use only)
list expected approval meeting # here ↑		for information		non-strategic		

Form: CR cover sheet, version 2 for 3GPP and SMG *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/CR-Form-v2.doc*

Proposed change affects: *(at least one should be marked with an X)*	(U)SIM	X	ME		UTRAN / Radio		Core Network	X

Source: Siemens Atea **Date:** 21-01-00

Subject: Authentication and key agreement

Work item: Security

Category: *(only one category shall be marked with an X)*

F	Correction	X	**Release:**	Phase 2	
A	Corresponds to a correction in an earlier release			Release 96	
B	Addition of feature			Release 97	
C	Functional modification of feature			Release 98	
D	Editorial modification			Release 99	X
				Release 00	

Reason for change: Bring TS 33.105 in line with the decisions taken in TSG SA and the CRs to TS 33.102 that have been approved. This includes the replacement of MODE by AMF, the change as regards input parameters to the computation of the re-synchronisation token AUTS, the deletion of the alternative mechanism (annex B) and the deletion of annex C on unspecified values, as all these value have in the mean time been specified.

Clauses affected: 5.1, annex B, annex C

Other specs affected:

Other 3G core specifications		→ List of CRs:
Other GSM core specifications		→ List of CRs:
MS test specifications		→ List of CRs:
BSS test specifications		→ List of CRs:
O&M specifications		→ List of CRs:

Other comments:

help.doc

<--------- double-click here for help and instructions on how to create a CR.

5.1 Authentication and key agreement

5.1.1 Overview

The mechanism for authentication and key agreement described in clause 6.3 of [1] requires the following cryptographic functions:

f0	the random challenge generating function;
f1	the network authentication function;
f1*	the re-synchronisation message authentication function;
f2	the user authentication function;
f3	the cipher key derivation function;
f4	the integrity key derivation function;
f5	the anonymity key derivation function.

5.1.1.1 Generation of quintets in the AuC

To generate a quintet the HLR/AuC

- computes a message authentication code for authentication MAC-A = $f1_K$(SQN || RAND || AMF), an expected response XRES = $f2_K$ (RAND), a cipher key CK = $f3_K$ (RAND) and an integrity key IK = $f4_K$ (RAND) where f4 is a key generating function.
- If SQN is to be concealed, in addition the HLR/AuC computes an anonymity key AK = $f5_K$ (RAND) and computes the concealed sequence number SQN ⊕ AK = SQN xor AK. Concealment of the sequence number is optional.
- Finally, the HLR/AuC assembles the authentication token AUTN = SQN [⊕ AK] || AMF || MAC-A and the quintet Q = (RAND, XRES, CK, IK, AUTN).

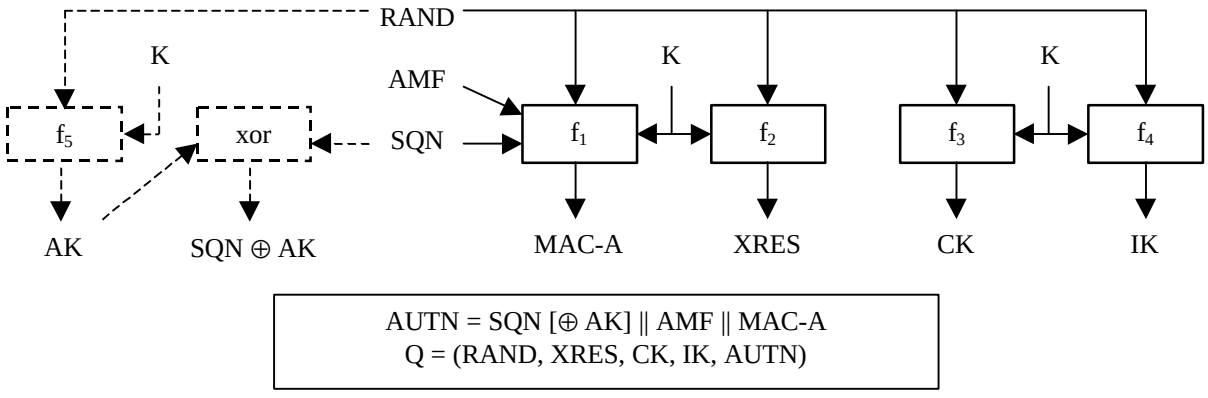

Figure 1: Generation of quintets in the AuC

5.1.1.2 Authentication and key derivation in the USIM

Upon receipt of a (RAND, AUTN) pair the USIM acts as follows:

- If the sequence number is concealed, the USIM computes the anonymity key AK = $f5_K$(RAND) and retrieves the unconcealed sequence number SQN = (SQN ⊕ AK) xor AK.

The USIM computes XMAC-A = $f1_K$ (SQN || RAND || AMF), the response RES = $f2_K$(RAND), the cipher key CK

= $f3_K$ (RAND) and the integrity key IK = $f4_K$ (RAND).

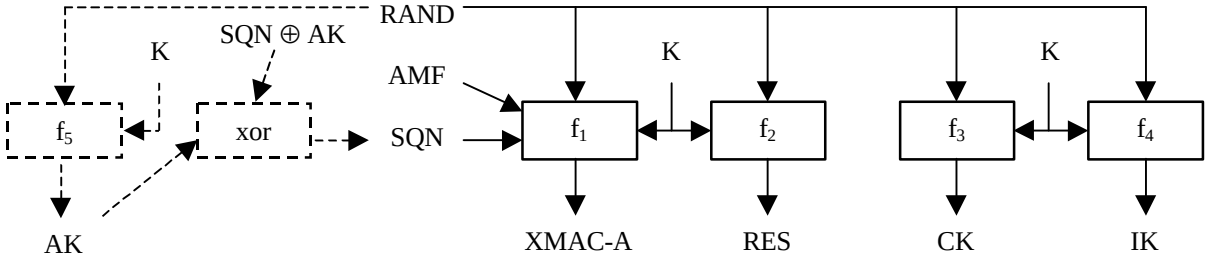

Figure 2: Authentication and key derivation in the USIM

5.1.1.3 Generation of re-synchronisation token in the USIM

Upon the assertion of a synchronisation failure, the USIM generates a re-synchronisation token as follows:

a) The USIM computes MAC-S = $f1^*_K(SQN_{MS} \| RAND \| AMF^*)$, whereby AMF* is a default value for AMF used in re-synchronisation.

b) If SQN_{MS} is to be concealed with an anonymity key AK, the USIM computes AK = $f5_K$(MAC-S || 0…0), whereby MAC-S forms the 12 most significant octets and 32 zeros form the 4 least significant octets of the required 16 octet input parameter, and the concealed counter value is then computed as $SQN_{MS} \oplus AK$.

c) The re-synchronisation token is constructed as AUTS = SQN_{MS} [⊕ AK] || MAC-S.

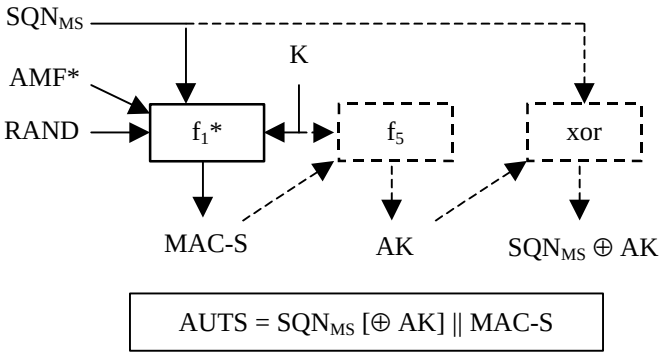

Figure 3: Generation of re-synchronisation token in the USIM

5.1.1.4 Re-synchronisation in the HLR/AuC

Upon receipt of an indication of synchronisation failure and a (AUTS, RAND) pair, the HLR/AuC may perform the following cryptographic functions:

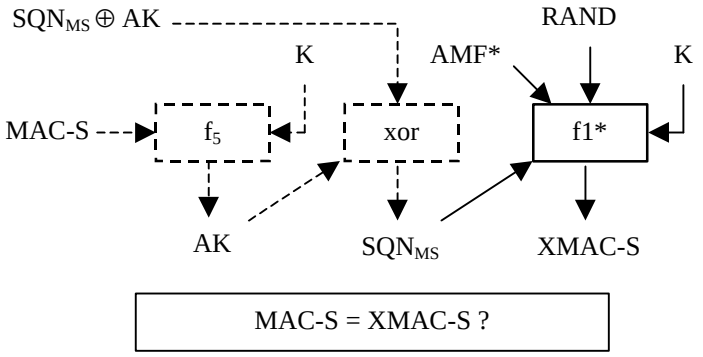

Figure 4: Re-synchronisation in the HLR/AuC

a) If SQN_{MS} is concealed with an anonymity key AK, the HLR/AuC computes AK = $f5_K$(MAC-S || 0…0), whereby MAC-S forms the 12 most significant octets and 32 zeros form the 4 least significant octets of the required 16 octet input parameter and retrieves the unconcealed counter value as SQN_{MS} = (SQN_{MS} ⊕ AK) xor AK.

b) If SQN generated from SQN_{HE} would not be acceptable, then the HLR/AuC computes XMAC-S = $f1^*_K$(SQN_{MS} || RAND || AMF*), whereby AMF* is a default value for AMF used in re-synchronisation.

5.1.2 Use

The functions f0—f5 shall only be used to provide mutual entity authentication between USIM and AuC, derive keys to protect user and signalling data transmitted over the radio access link and conceal the sequence number to protect user identity confidentiality. The function f1* shall only be used to provide data origin authentication for the synchronisation failure information sent by the USIM to the AuC.

5.1.3 Allocation

The functions f1—f5 and f1*are allocated to the Authentication Centre (AuC) and the USIM. The function f0 is allocated to the AuC.

5.1.4 Extent of standardisation

The functions f0—f5 and f1*are proprietary to the home environment. Examples of the functions f1, f1* and f2 are CBC-MACs or H-MACs [3].

5.1.5 Implementation and operational considerations

The functions f1—f5 and f1* shall be designed so that they can be implemented on an IC card equipped with a 8-bit microprocessor running at 3.25 MHz with 8 kbyte ROM and 300byte RAM and produce AK, XMAC-A, RES, CK and IK in less than 500 ms execution time.

5.1.6 Type of algorithm

5.1.6.1 f0

f0: the random challenge generating function

f0: (internal state) → RAND

f0 should be (pseudo) random number generating function.

5.1.6.2 f1

f1: the network authentication function

f1: (K; SQN, RAND, AMF) → MAC-A (or XMAC-A)

f1 should be a MAC function. In particular, it shall be computationally infeasible to derive K from knowledge of RAND, SQN, AMF and MAC-A (or XMAC-A).

5.1.6.3 f1*

f1*: the re-synchronisation message authentication function

f1*: (K; SQN, RAND, AMF) → MAC-S (or XMAC-S)

f1 should be a MAC function. In particular, it shall be computationally infeasible to derive K from knowledge of RAND, SQN, AMF and MAC-S (or XMAC-S).

5.1.6.4 f2

f2: the user authentication function

f2: (K; RAND) → RES (or XRES)

f2 should be a MAC function. In particular, it shall be computationally infeasible to derive K from knowledge of RAND and RES (or XRES).

5.1.6.5 f3

f3: the cipher key derivation function

f3: (K; RAND) → CK

f3 should be a key derivation function. In particular, it shall be computationally infeasible to derive K from knowledge of RAND and CK.

5.1.6.6 f4

f4: the integrity key derivation function

f4: (K; RAND) → IK

f4 should be a key derivation function. In particular, it shall be computationally infeasible to derive K from knowledge of RAND and IK.

5.1.6.7 f5

f5: the anonymity key derivation function

f5: (K; RAND) → AK

f5 should be a key derivation function. In particular, it shall be computationally infeasible to derive K from knowledge of RAND and AK.

The use of f5 is optional.

5.1.7 Interface

5.1.7.1 K

K: the subscriber authentication key

K[0], K[1], …, K[127]

The length of K is 128 bits. The subscriber authentication key K is a long term secret key stored in the USIM and the AuC.

5.1.7.2 RAND

RAND: the random challenge

RAND[0], RAND[1], …, RAND[127]

The length of RAND is 128 bits.

5.1.7.3 SQN

SQN: the sequence number

SQN[0], SQN[1], …, SQN[47]

The length of SQN is 48 bits. The AuC should include a fresh sequence number in each authentication token. The verification of the freshness of the sequence number by the USIM constitutes to entity authentication of the network to the user.

5.1.7.4 AMF

AMF: the authentication management field

AMF[0], AMF[1], …, AMF[15]

The length of AMF is 16 bits. The use of AMF is not standardised. Example uses of the AMF are provided in annex F of TS 33.102.

5.1.7.6 MAC-A (equivalent for XMAC-A)

MAC-A: the message authentication code used for authentication of the network to the user

MAC-A[0], MAC-A[1], …, MAC-A[63]

The length of MAC-A is 64 bits. MAC-A authenticates the data integrity and the data origin of RAND, SQN and AMF. The verification of MAC-A by the USIM constitutes to entity authentication of the network to the user.

5.1.7.7 MAC-S (equivalent for XMAC-S)

MAC-S: the message authentication code used to provide data origin authentication for the synchronisation failure information sent by the USIM to the AuC.

MAC-S[0], MAC-S[1], …, MAC-S[63]

The length of MAC-S is 64 bits. MAC-S authenticates the data integrity and the data origin of RAND, SQN and AMF. MAC-S is generated by the USIM and verified by the AuC.

5.1.7.8 RES (or XRES)

RES: the user response

RES[0], RES[1], …, RES[31…127]

The maximum length of RES and XRES is 128 bits and the minimum is 32 bits. RES and XRES constitute to entity authentication of the user to the network.

5.1.7.9 CK

CK: the cipher key

CK[0], CK[1], …, CK[127]

The length of CK is 128 bits. In case the effective key length should need to be made smaller than 128 bits, the most significant bits of CK shall carry the effective key information, whereas the remaining, least significant bits shall be set zero.

5.1.7.10 IK

IK: the integrity key

IK[0], IK[1], …, IK[127]

The length of IK is 128 bits. In case the effective key length should need to be made smaller than 128 bits, the most significant bits of IK shall carry the effective key information, whereas the remaining, least significant bits shall be set zero.

5.1.7.11 AK

AK: the anonymity key

AK[0], AK[1], …, AK[47]

The length of AK is 48 bits. It equals the length of SQN.

[1] RSn and/or RSu can be a random number or a counter