3GPP TSG SA WG3 Security — SA3#33 **S3-040297**
10-14 May 2004, Beijing, China

CR-Form-v7

CHANGE REQUEST

⌘	**33.105**	**CR**	**CRNum**	⌘ **rev**	**-**	⌘	Current version:	**4.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [X] ME [ ] Radio Access Network [ ] Core Network [X]

Title:	⌘	Correction of inconsistencies in AK computation for re-synchronisation		
Source:	⌘	Orange		
Work item code:	⌘	UTRAN Security	***Date:*** ⌘	23/04/2004
Category:	⌘	**F**	***Release:*** ⌘	*Rel-4*
		Use one of the following categories: ***F** (correction)* ***A** (corresponds to a correction in an earlier release)* ***B** (addition of feature),* ***C** (functional modification of feature)* ***D** (editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *Rel-4 (Release 4)* *Rel-5 (Release 5)* *Rel-6 (Release 6)*

Reason for change:	⌘	f5 is used instead of f5* in figures 3 and 4.
Summary of change:	⌘	f5 is replaced by f5* in figures 3 and 4.
Consequences if not approved:	⌘	Consistency problem. Potential misinterpretation of AK computation for re-synchronisation.

Clauses affected:	⌘	5.1.1.3, 5.1.1.4				
		Y	**N**			
Other specs affected:	⌘			Other core specifications	⌘	
				Test specifications		
				O&M Specifications		
Other comments:	⌘					

****************** BEGIN OF CHANGE ******************

5 Functional algorithm requirements

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 for normal operation;
f5*	the anonymity key derivation function for re-synchronisation.

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).

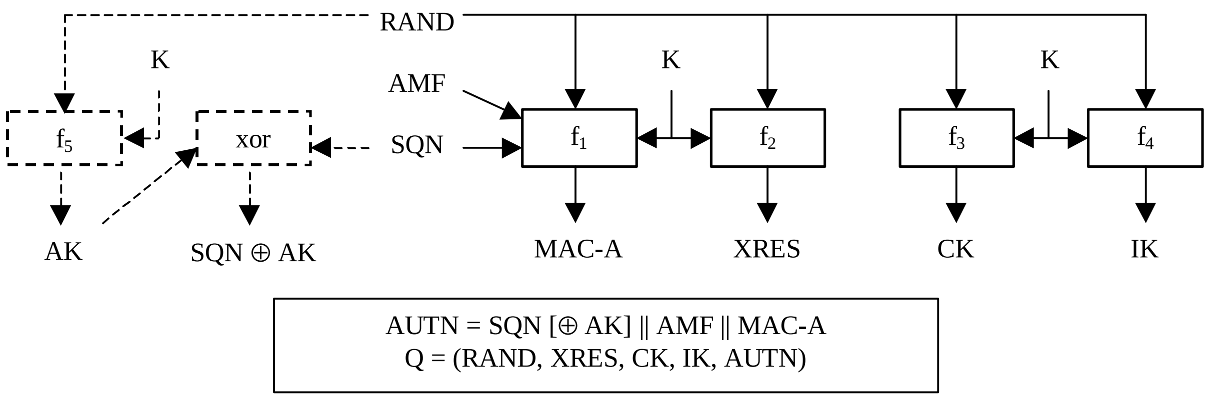

Figure 1: Generation of quintets in the AuC

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 $SQN = (SQN \oplus AK)$ xor AK.

  The USIM computes $XMAC\text{-}A = f1_K (SQN \| RAND \| AMF)$, the response $RES = f2_K(RAND)$, the cipher key $CK = f3_K (RAND)$ and the integrity key $IK = f4_K (RAND)$.

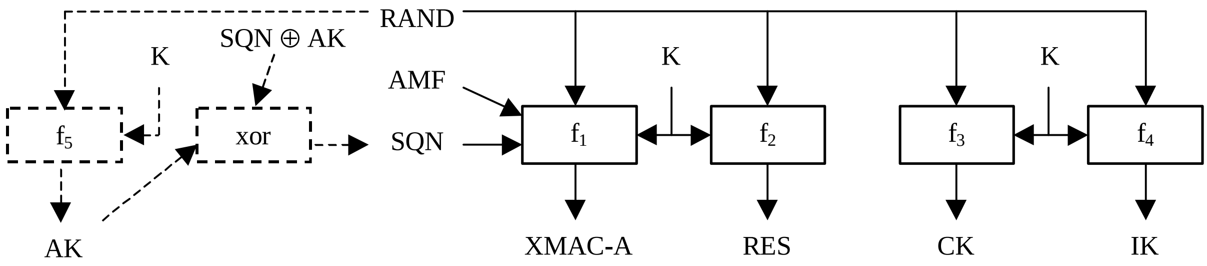

Figure 2: Authentication and key derivation in the USIM

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 $MAC\text{-}S = f1^*_K(SQN_{MS} \| RAND \| AMF^*)$, 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 $AK = f5^*_K(RAND)$, and the concealed counter value is then computed as $SQN_{MS} \oplus AK$.

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

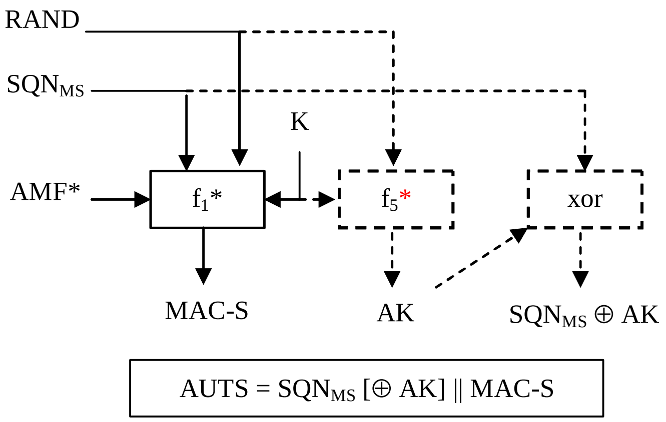

Figure 3: Generation of re-synchronisation token in the USIM

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:

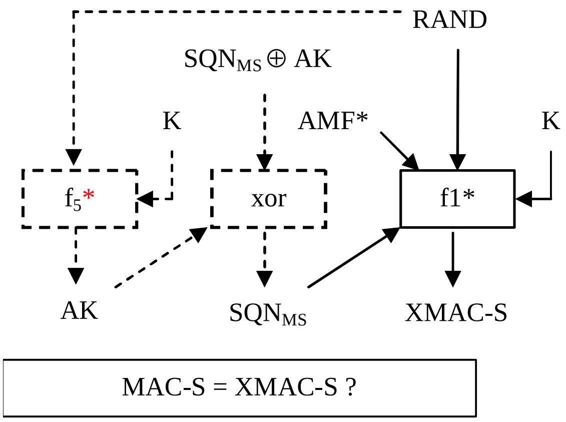

Figure 4: Re-synchronisation in the HLR/AuC

a) If SQN_{MS} is concealed with an anonymity key AK, the HLR/AuC computes AK = $f5*_K$(RAND)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.

****************** END OF CHANGE ******************