3GPP TSG SA WG3 Security — S3#20 **S3-010535**

16-19 October 2001, Sidney, Australia

CR-Form-v4

CHANGE REQUEST

⌘ **33.102** **CR** ⌘ ev **-** ⌘ Current version: **4.2.0** ⌘

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

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Sequence Number Management Corrections
Source:	⌘	Siemens Atea
Work item code:	⌘	Security
Date:	⌘	8 October 2001
Category:	⌘	**A**
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)*

Reason for change:	⌘	Clarifications in order to make implicit requirements explicit and correction of editorial mistakes on sequence number management in Annex C.
Summary of change:	⌘	1) Clarifications for Sequence Number acceptance in Annex C.2. 2) Smaller editorial corrections in Annex C
Consequences if not approved:	⌘	Possible misinterpretation of the examples given in Annex C

Clauses affected:	⌘	Annex C		
Other specs affected:	⌘	[X] Other core specifications	⌘	TS 31.102
		[ ] Test specifications		
		[ ] O&M Specifications		
Other comments:	⌘			

Annex C (informative): Management of sequence numbers

This annex is devoted to the management of sequence numbers for the authentication and key agreement protocol.

C.1 Generation of sequence numbers in the Authentication Centre

C.1.1 Sequence number generation schemes

C.1.1.1 General scheme

According to section 6.3 of this specification, authentication vectors are generated in the authentication centre (AuC) using sequence numbers. This section specifies how these sequence numbers are generated. Authentication vectors may be generated and sent by the AuC in batches. The sequence numbers for the authentication vectors in a batch are generated one after the other according to the process described below.

(1) In its binary representation, the sequence number consists of two concatenated parts $SQN = SEQ \,||\, IND$.
IND is an index used in the array scheme described in C.1.2 and C.2.2. *SEQ* in its turn consists of two concatenated parts $SEQ = SEQ1 \,||\, SEQ2$. *SEQ1* represents the most significant bits of *SEQ*, and *SEQ2* represents the least significant bits of *SEQ*. *IND* represents the least significant bits of *SQN*.

(2) There is a counter SQN_{HE} in the HE. *SQN* is stored by this counter. SQN_{HE} is an individual counter, i.e. there is one per user. We have $SQN_{HE} = SEQ_{HE} \,||\, IND_{HE}$.

(3) There is a global counter, e.g. a clock giving universal time. For short we call the value of this global counter at any one time *GLC*. If *GLC* is taken from a clock it is computed mod p, where $p = 2^n$ and n is the length of *GLC* and of *SEQ2* in bits.

(4) If *GLC* is taken from a clock then there is a number $D > 0$ such that the following holds:
(i) the time interval between two consecutive increases of the clock (the clock unit) shall be chosen such that, for each user, at most *D* batches are generated at the AuC during any *D* clock units;
(ii) the clock rate shall be significantly higher than the average rate at which batches are generated for any user;
(iii) $D << 2^n$.

(5) When the HE needs new sequence numbers *SQN* to create a new batch of authentication vectors, HE retrieves the (user-specific) value of $SEQ_{HE} = SEQ1_{HE} \,||\, SEQ2_{HE}$ from the database.
(i) If $SEQ2_{HE} < GLC < SEQ2_{HE} + p - D + 1$ then HE sets $SEQ = SEQ1_{HE} \,||\, GLC$;
(ii) if $GLC \leq SEQ2_{HE} \leq GLC+D - 1$ or $SEQ2_{HE} + p - D + 1 \leq GLC$ then HE sets $SEQ = SEQ_{HE} + 1$;
(iii) if $GLC+D - 1 < SEQ2_{HE}$ then HE sets $SEQ = (SEQ1_{HE} + 1) \,||\, GLC$.
(iv) After the generation of the authentication vector has been completed SEQ_{HE} is reset to *SEQ;*
(v) for the handling of *IND* see C.1.2.

NOTES

1. The clock unit and the value *D* have to be chosen with care so that condition (4)(i) is satisfied for every user at all times. Otherwise, user identity confidentiality may be compromised. When the

parameters are chosen appropriately sequence numbers for a particular user do not reveal significant information about the user's identity.
If authentication vectors for the CS and the PS domains are not separated by other means it is recommended to choose D >1 as requests from the two different domains may arrive completely independently.

2. By setting the parameters in C.1.1.1 (1) to (5) in an appropriate way the general scheme specified in this subsection also includes the cases where either SEQ2 is void and SEQ = SEQ1 or else, SEQ1 is void and SEQ = SEQ2, as follows:
(a) If SEQ2 is void the generation of sequence numbers is not time-based. We then formally set SEQ2 ≡ GLC ≡ 0 (identical to zero) and D = 1. Conditions (4)(i) to (iii) do not apply as there is no clock. Then (5)(ii) always holds, and SEQ is incremented by 1 at each request. For better readability, this case is separated out in C.1.1.2.
(b) If SEQ1 is void then we set D = 1. Assuming a start condition $SEQ2_{HE}$ < GLC and the absence of failures in the AuC, the condition (5)(i) then always holds, and SEQ = GLC for each request, i.e. the generation of sequence numbers is entirely time-based. In order to also accommodate potential failures in the AuC for entirely time-based sequence number , the variant described in the following Annex C.1.1.3 may be used.

C.1.1.2 Generation of sequence numbers which are not time-based

The HE/AuC shall maintain a counter for each user, $SQN_{HE} = SEQ_{HE} \| IND_{HE}$. To generate a fresh sequence number, SEQ_{HE} is incremented by 1, and the new counter value is used to generate the next authentication vector. For the handling of *IND* see C.1.2.

C.1.1.3 Time-based sequence number generation

In its binary representation, the sequence number consists of two concatenated parts $SQN = SEQ \| IND$. The part *SEQ* is not divided into two parts. The global counter *GLC* is thus as long as *SEQ*. Instead of storing the individual counter SEQ_{HE} in the HE there is a value *DIF* stored in the HE which is individual for each user. The *DIF* value represents the current difference between generated *SEQ* values for that user and the *GLC*.

When the HE needs new sequence numbers *SQN* to create new authentication vectors, HE retrieves the (user-specific) value of *DIF* from the data base and calculates *SEQ* values as $SEQ = GLC + DIF$.

The *DIF* value may have to be updated in the HE only during the re-synchronization procedure. In this case the DIF value is set as $DIF = SEQ_{MS} - GLC$ where $SQN_{MS} = SEQ_{MS} \| IND_{MS}$ is the value sent by USIM in the re-synchronization procedure.

C.1.2 Support for the array mechanism

This subsection applies to all three schemes presented in subsection C.1.1.

Each time an authentication vector is generated, the AuC shall retrieve IND_{HE} from storage and allocate a new index value *IND* for that vector according to suitable rules and include it in the appropriate part of *SQN*. The index value may range from 0 to a -1 where a is the size of the array.

An example value for the array size a is given in Annex C.3.
The exact rules for index allocation are left unspecified. Guidelines are given in Annex C.3.4.

C.2 Handling of sequence numbers in the USIM

This section assumes that sequence numbers are generated according to Annex C.1.

The USIM keeps track of an array of sequence number values it has accepted. Let $SQN_{MS} = SEQ_{MS} || IND_{MS}$ denote the highest sequence number in the array.

C.2.1 Protection against wrap around of counter in the USIM

The USIM will not accept arbitrary jumps in sequence numbers, but only increases by a value of at most Δ.

Therefore (before applying the freshness conditions of Annex C.2.2) the received sequence number SQN shall only be accepted by the USIM if $SEQ\text{-}SEQ_{MS} \leq \Delta$. If SQN can not be accepted then the USIM shall generate a synchronisation failure message using SQN_{MS}.

Conditions on the choice of Δ:

(1) Δ shall be sufficiently large so that the MS will not receive any sequence number with $SEQ - SEQ_{MS} \geq \Delta$ if the HE/AuC functions correctly.

(2) In order to prevent that SEQ_{MS} ever reaches the maximum batch number value *SEQmax* during the lifetime of the USIM the minimum number of steps $SEQmax / \Delta$ required to reach *SEQmax* shall be sufficiently large.

C.2.2 Verification of sequence number freshness in the USIM

The USIM shall maintain an array of *a* previously accepted sequence number components: SEQ_{MS} (0), SEQ_{MS} (1),… SEQ_{MS} (*a*-1). The initial sequence number value in each array element shall be zero.

To verify that the received sequence number *SQN* is fresh, the USIM shall compare the received *SQN* with the sequence number in the array element indexed using the index value *IND* contained in *SQN*, i.e. with the array entry SEQ_{MS} (i) where i = *IND* is the index value.

(a) If $SEQ > SEQ_{MS}$ (i) the USIM shall consider the sequence number to be guaranteed fresh and subsequently shall set SEQ_{MS} (i) to *SEQ*.

(b) If $SEQ \leq SEQ_{MS}$ (i) the USIM shall generate a synchronisation failure message using the highest previously accepted sequence number anywhere in the array, i.e. SQN_{MS}.

The USIM shall also be able to put a limit *L* on the difference between SEQ*SQN*$_{MS}$ and a received sequence number *component SEQSQN*. If such a limit L is applied then, before verifying the above conditions (a) and (b), the sequence number shall only be accepted by the USIM if $SEQQN_{MS} - SEQQN < L$. If SQN can not be accepted then the USIM shall generate a synchronisation failure message using SQN_{MS}.

C.2.3 Notes

1. Using the above array mechanism, it is not required that a previously visited VLR/SGSN deletes the unused authentication vectors when a user de-registers from the serving network (super-charger

concept). Retaining the authentication vectors for use when the user returns later may be more efficient as regards signalling when a user abroad switches a lot between two serving networks.

2. The array mechanism may also be used to avoid unjustified rejection of user authentication requests when authentication vectors in two VLR/SGSNs from different mobility management domains (circuit and packet) are used in an interleaving fashion.

3. When a VLR/SGSN uses fresh authentication vectors obtained during a previous visit of the user, the USIM can reject them although they have not been used before (because the array size *a* and the age limit *L* are finite). Rejection of a sequence number can therefore occur in normal operation, i.e., it is not necessarily caused by (malicious) replay or a database failure.

4. The mechanism presented in this section may allow the USIM to exploit knowledge about which authentication vectors were sent to the same VLR/SGSN. It may be assumed that authentication vectors sent to the same VLR/SGSN are always used in the correct order. Consequently, only one sequence number among those sent to the same VLR/SGSN has to be stored.

5. With the exception of SQN_{MS}, the entries of the array need not be stored in full length if a limit *L* (age limit) on the difference between SEQSQN_{MS} and an received~~accepted~~ sequence number *component SEQ* is applied.

6. Condition (2) of Annex C.2.1 on Δ means that SQN_{MS} can reach its maximum value only after a minimum of *SEQ~~QN~~max /Δ* successful authentications have taken place.

7. There is a dependency of the choice of Δ and the size *n* of global counter GLC in Annex C.1.1.1: Δ shall be chosen larger than 2^n.

C.3 Sequence number management profiles

This section provides examples how values for the parameters defined in sections C.1 and C.2 may be chosen in a coherent way. These examples may serve as references when specifying practical sequence number management schemes. There is one example set of values for each of the three types of sequence number generation schemes:

- partly time-based corresponding to Annex C.1.1.1;
- not time-based corresponding to Annex C.1.1.2;
- entirely time-based corresponding to Annex C.1.1.3.

C.3.1 Profile 1: management of sequence numbers which are partly time-based

Generation of sequence numbers:

This follows the general scheme for the generation of sequence numbers specified in Annex C.1.1.1. The following parameter values are suggested for reference:

Time unit of the clock: 1 second

Length of IND in bits = 5.

Length of SEQ~~Q~~N2 in bits = n : 24
This means that GLC will wrap around after $p = 2^n = 2^{24}$ seconds = 194 days. This ensures that most users

will have become active at least once during this period.
This implies a length of SEQ1 in bits = 19.

Start conditions: Choose SQN_{HE} = 0 for all users and GLC = 1.

Arrival rate temporarily higher than clock rate: Choose $D = 2^{16}$.
D may be chosen quite large as long as the conditions in C.1.1.1 (4)(ii) and (iii) are satisfied. Choosing $D = 2^{16} = 65536$ means that the condition in C.1.1.1 (4)(i) is satisfied unless more than 65536 requests for batches arrive within over 18 hours which is practically impossible.

Verification of sequence numbers in the USIM:

This follows the handling of sequence numbers in the USIM specified in Annex C.2.

Length of the array: a = 32.
This satisfies the requirement in section 6.3.2 that the mechanism for the verification of sequence numbers shall ensure that a sequence number can still be accepted if it is among the last x sequence numbers generated.

Protection against wrap around: Choose $\Delta = 2^{28}$.
Choosing $\Delta = 2^{28}$ means that an attack to force the counter in the USIM to wrap around would require at least $SEQmax/\Delta = 2^{15} > 32.000$ successful authentications (cf. note 6 of C.2.34). We have $\Delta > p$, as required in note 7 of C.2.3.

Age limit for sequence numbers:
The use of such a limit is optional. The choice of a value for the parameter L affects only the USIM. It has no impact on the choice of other parameters and it entirely up to the operator, depending on his security policy. Therefore no particular value is suggested here. To give an example: if the policy stipulates that authentication vectors older than x seconds shall be rejected then L has to be set to x as the time unit of the clock is 1 second.

User anonymity: the value of SQN does not allow to trace the user over longer periods. Therefore, there may be no need to conceal SQN by an anonymity key as specified in section 6.3.

C.3.2 Profile 2: management of sequence numbers which are not time-based

Generation of sequence numbers:

This follows the scheme for the generation of sequence numbers specified in Annex C.1.1.2. The following parameter values are suggested for reference:

Length of IND in bits = 5.

Start conditions: SQN_{HE} = 0 for all users.

Verification of sequence numbers in the USIM:

Length of the array: a = 32

Protection against wrap around: Choose $\Delta = 2^{28}$.
Choosing $\Delta = 2^{28}$ means that an attack to force the counter in the USIM to wrap around would require at least $SEQmax/\Delta = 2^{15} > 32.000$ successful authentications (cf. note 6 of C.2.34). Note 7 of Annex C.2.3 does not apply.

Age limit for sequence numbers:
There is no clock here. So, the "age" limit would be interpreted as the maximum allowed difference

between SQN_{MS} (see section 6.3) and the sequence number received. The use of such a limit is optional. The choice of a value for the parameter L affects only the USIM. It has no impact on the choice of other parameters and it entirely up to the operator, depending on his security policy. Therefore no particular value is suggested here.

User anonymity: the value of SQN may allow to trace the user over longer periods. If this is a concern then SQN has to be concealed by an anonymity key as specified in section 6.3.

C.3.3 Profile 3: management of sequence numbers which are entirely time-based

Generation of sequence numbers:

This follows the scheme for the generation of sequence numbers specified in Annex C.1.1.3. The following parameter values are suggested for reference:

Time unit of the clock: It has to be chosen in such a way that no two requests for a batch of authentication vectors arrive during one time unit. Value = 0.1 seconds

Length of IND in bits = 5.

Start conditions: GLC = 1 and, for all users, DIF = 0.

Verification of sequence numbers in the USIM:

This is done according to the handling of sequence numbers in the USIM specified in Annex C.2.

Length of the array: a = 32.
This satisfies the requirement in section 6.3.2 that the mechanism for the verification of sequence numbers shall ensure that a sequence number can still be accepted if it is among the last x sequence numbers generated.

Protection against wrap around: Choose $\Delta = 2^{28}$.
Choosing $\Delta = 2^{28}$ means that an attack to force the counter in the USIM to wrap around would require at least $SEQmax/\Delta = 2^{15} > 32.000$ successful authentications (cf. note 6 of C.2.34). Note 7 of C.2.34 does not apply.

Age limit for sequence numbers:
The use of such a limit is optional. The choice of a value for the parameter L affects only the USIM. It has no impact on the choice of other parameters and it entirely up to the operator, depending on his security policy. Therefore no particular value is suggested here. To give an example: if the policy stipulates that authentication vectors older than x time units shall be rejected then L has to be set to x.

User anonymity: the value of SQN does not allow to trace the user over longer periods. Therefore, there may be no need to conceal SQN by an anonymity key as specified in section 6.3.

C.3.4 Guidelines for the allocation of the index values in the array scheme

- **General rule**: index values *IND* used in the array scheme, according to Annex C.1.2, shall be allocated cyclically within its range 0, ... , *a-1*. This means that the index value *IND* used with the previously generated authentication vector is stored in SQN_{HE} , and the next authentication vector shall use index value *IND +1 mod a*.

It may be useful to allow exceptions to this general rule when additional information is available. This includes:

- Authentication vectors distributed within the same batch shall have the same index value.

  The Authentication Data Request MAP message contains information about the domain type (CS or PS) of the requesting serving node from which the request originates. It is recommended to use this information in the following way. Support for this use is, however, not required for an implementation to claim compliance to Annex C.

- Authentication vectors distributed to different service domains shall have different index values (i.e. separate ranges of index values are reserved for PS and CS operation).

In future releases there may be additional information about the requesting node identity. If this information is available it is recommended to use it in the following way:

- If the new request comes from the same serving node as the previous request, then the index value used for the new request shall be the same as was used for the previous request.

C.4 Guidelines for interoperability in a multi-vendor environment

The specification of a sequence number management scheme affects only the USIM and the AuC which are both under the control of one operator. Therefore, the specification of such a scheme is entirely at the discretion of an operator. Nevertheless, certain operators may not want to define a scheme of their own. Instead, they may want to rely on vendors implementing one of the schemes according to the profiles in C.3 or variants thereof. If these operators have multiple vendors for USIMs and/or AuCs, and the operators wish to move subscribers from the AuC of one vendor to that supplied by another one implementing a different scheme then this will work smoothly only when the following guidelines are adhered to by all the sequence number management schemes implemented in the operator's domain.

- The array mechanism specified in C.1.2 and C.2 is used in the USIM to verify SQNs.

- Relation to Annex F: if the AMF field is used to signal further parameters relevant to sequence number management (age limit L) then the formats of the AMF and its interpretation by the USIM must be the same for all implementations in the operator's domain.

- Δ is larger than a specified minimum.
  This is necessary to accommodate schemes as in C.3.2 according to note 7 of C.2.3.
  We propose $\Delta \geq 2^{28}$.

- There are no requirements on the synchronicity of clocks in different AuCs for the time-based schemes. For the entirely time-based scheme, the following is recommended when moving users from one AuC to another one: The DIF value is updated in an appropriate manner when moving subscribers from an AuC to another AuC. More specifically, assume a user is moved from AuC1 to AuC2. If AuC1 is of profile 3 and AuC2 is of any profile then AuC1 sends GLC+DIF as SEQ_HE

to AuC2. In the receiving end, if AuC2 is of profile 3 while AuC1 is of any profile then AuC2 sets DIF value for this user as DIF = SEQ_HE - GLC.