3GPP TSG-RAN meeting #4
Miami, USA, 17-18 June 1999

Document RP-99333

3G CHANGE REQUEST

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

TS 25.301 **CR** **003** Current Version: 3.0.1

3G specification number ↑ *↑ CR number as allocated by 3G support team*

For submission to TSG RAN#4 *list TSG meeting no. here ↑* for approval [X] for information [ ] *(only one box should be marked with an X)*

Form: 3G CR cover sheet, version 1.0 *The latest version of this form is available from: ftp://ftp.3gpp.org/Information/3GCRF-xx.rtf*

Proposed change affects: *(at least one should be marked with an X)* USIM [ ] ME [X] UTRAN [X] Core Network [ ]

Source: TSG-RAN WG2 **Date:** 08/06/99

Subject: Description of ciphering model

3G Work item:

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

F	Correction	[ ]
A	Corresponds to a correction in a 2G specification	[ ]
B	Addition of feature	[ ]
C	Functional modification of feature	[X]
D	Editorial modification	[ ]

Reason for change: The ciphering model was not yet described in TS 25.301. This CR is proposing a model according to the agreement reached at RAN2#4 meeting.

Clauses affected:

Other specs affected:

Other 3G core specifications	[ ]	→ List of CRs:
Other 2G 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.

5.3.1.2 MAC functions

The functions of MAC include:

- **Mapping between logical channels and transport channels.** The MAC is responsible for mapping of logical channel(s) onto the appropriate transport channel(s).
- **Selection of appropriate Transport Format for each Transport Channel depending on instantaneous source rate**. Given the Transport Format Combination Set assigned by RRC, MAC selects the appropriate transport format within an assigned transport format set for each active transport channel depending on source rate**.** The control of transport formats ensures efficient use of transport channels.
- **Priority handling between data flows of one UE**. When selecting between the Transport Format Combinations in the given Transport Format Combination Set, priorities of the data flows to be mapped onto the corresponding Transport Channels can be taken into account. Priorities are e.g. given by attributes of radio access bearer services and RLC buffer status. The priority handling is achieved by selecting a Transport Format Combination for which high priority data is mapped onto L1 with a "high bit rate" Transport Format, at the same time letting lower priority data be mapped with a "low bit rate" (could be zero bit rate) Transport Format. Transport format selection may also take into account transmit power indication from Layer 1.
- **Priority handling between UEs by means of dynamic scheduling**. In order to utilize the spectrum resources efficiently for bursty transfer, a dynamic scheduling function may be applied. Priority handling on common and shared transport channels is realized by MAC. Note that for dedicated transport channels, the equivalent of the dynamic scheduling function is implicitly included as part of the reconfiguration function of the RRC sublayer. For TDD it is regarded as further study item.

  Note that in the TDD mode the data to be transported are represented in terms of sets of resource units.
- **Scheduling of broadcast, paging and notification messages**. This function provides mechanisms for efficient transfer of broadcast, paging and notification messages by means of appropriate scheduling and repetition of the messages.
- **Identification of UEs on common transport channels**. When a particular UE is addressed on a common downlink channel, or when a UE is using the RACH, there is a need for inband identification of the UE. Since the MAC layer handles the access to, and multiplexing onto, the transport channels, the identification functionality is naturally also placed in MAC.
- **Multiplexing/demultiplexing of higher layer PDUs into/from transport blocks delivered to/from the physical layer on common transport channels**. MAC should support service multiplexing for common transport channels, since the physical layer does not support multiplexing of these channels.
- **Multiplexing/demultiplexing of higher layer PDUs into/from transport block sets delivered to/from the physical layer on dedicated transport channels**. The MAC allows service multiplexing for dedicated transport channels. This function can be utilized when several upper layer services (e.g. RLC instances) can be mapped efficiently on the same transport channel. In this case the identification of multiplexing is contained in the MAC protocol control information.
- **Traffic volume monitoring**. Measurement of traffic volume on logical channels and reporting to RRC. Based on the reported traffic volume information, RRC performs transport channel switching decisions.
- **Routing of higher layer signalling**. This function performs the mapping of higher layer signalling messages to the appropriate transport channel. This function is required in TDD mode, where resource allocation is performed by the MAC autonomously.
- **Maintenance of a MAC signalling connection between peer MAC entities**. This function supports unacknowledged transfer of MAC-internal messages between peer MAC entities. A MAC signalling connection is required in the TDD mode.
- **Monitoring the links of the assigned resources**. This function provides means for monitoring link quality in TDD mode (used by MAC for fast DCA).
- **Dynamic Transport Channel type switching.** Execution of the switching between common and dedicated transport channels based on a switching decision derived by RRC**.**
- **Ciphering**. This function prevents unauthorised acquisition of data. Ciphering is performed in the MAC layer for transparent RLC mode.

The following potential functions are regarded as further study items:

- **Constrained execution of open loop power control algorithms**. This function establishes layer 1 power levels within the constraints of open loop power control set by RRC.

  [Note: Details of this function need to be clarified.]

- **Processing of messages received at common control channels**. This function is applied in TDD mode to support a data transfer on common control channels to support MAC operation (needed for fast DCA details are ffs.).

- **Successive Transmission on RACH.** When the mobile station continues to transmit the succeeding (second or more) radio frames because the message length is longer than a radio frame, the transmission timing offset, the RACH spreading code and signature shall be determined as follows: The transmission timing offset (frame and/or slot) shall be determined pseudo-randomly. The RACH spreading code and the signature of the succeeding radio frame can be determined pseudo-randomly. The same RNTI shall be used as in the previous radio frame (for the radio frames belonging to the same higher layer PDU).

  [Note: This function requires further clarification. Contributions are invited.]

~~- **Ciphering**. This function prevents unauthorised acquisition of data.~~

~~*[Note: Ciphering is considered as further study item. This includes consideration where it is applied, for instance on MAC, RLC, or elsewhere, cf. Sec. 8.]*~~

- **Access Service Class selection for RACH transmission**. The RACH resources (i.e. access slots and preamble signatures) may be divided between different Access Service Classes in order to provide different priorities of RACH usage. This function selects, based upon the type of data to be transmitted, the RACH parameters in accordance with the Service Access Class assignment.

  [Note: This function may support admission control. Its impact on BCCH capacity and its effects on RACH interference, retransmission and back-off time remains ffs.]

5.3.2.2 RLC Functions

- **Connection Control.** This function performs establishment, release, and maintenance of a RLC connection.
- **Segmentation and reassembly.** This function performs segmentation/reassembly of variable-length higher layer PDUs into/from smaller RLC Payload Units (PUs). One RLC PDU carries one PU or, in case header compression is applied several RLC PUs. The size oft the smallest retransmission unit shall be determined by the smallest possible bit rate. The RLC PDU size is adjustable to the actual set of transport formats.
- **Header compression**. The feature to include several Payload Units into one RLC PDU is referred to as RLC header compression. RLC header compression can be applied for acknowledged data transfer service. Its applicability shall be negotiable between network and UE. Application of RLC header compression is optional for the network but it shall be supported by the UE mandatory.
- **Concatenation.** If the contents of an RLC SDU does not fill an integer number of RLC PUs, the first segment of the next RLC SDU may be put into the RLC PU in concatenation with the last segment of the previous RLC SDU.
- **Padding.** When concatenation is not applicable and the remaining data to be transmitted does not fill an entire RLC PDU of given size, the remainder of the data field shall be filled with padding bits.
- **Transfer of user data.** This function is used for conveyance of data between users of RLC services. RLC supports acknowledged, unacknowledged and transparent data transfer. Transfer of user data is controlled by QoS setting.
- **Error correction.** This function provides error correction by retransmission (e.g. Selective Repeat, Go Back N, or a Stop-and-Wait ARQ) in acknowledged data transfer mode.
- **In-sequence delivery of higher layer PDUs**. This function preserves the order of higher layer PDUs that were submitted for transfer by RLC using the acknowledged data transfer service. If this function is not used, out-of-sequence delivery is provided.
- **Duplicate Detection.** This function detects duplicated received RLC PDUs and ensures that the resultant higher Layer PDU is delivered only once to the upper layer.
- **Flow control**. This function allows an RLC receiver to control the rate at which the peer RLC transmitting entity may send information.
- **Sequence number check (Unacknowledged data transfer mode).** This function guarantees the integrity of reassembled PDUs and provides a mechanism for the detection of corrupted RLC SDUs through checking sequence number in RLC PDUs when they are reassembled into a RLC SDU. A corrupted RLC SDU will be discarded.
- **Protocol error detection and recovery**. This function detects and recovers from errors in the operation of the RLC protocol.
- **Ciphering**. This function prevents unauthorised acquisition of data. Ciphering is performed in RLC layer for non-transparent RLC mode.

The following potential function(s) are regarded as further study items:

- **Suspend/resume function**. Suspension and resumption of data transfer as in e.g. LAPDm (cf. GSM 04.05).

~~- **Ciphering**. This function prevents unauthorised acquisition of data.~~

- **Quick repeat (C plane only).** This function provides mechanisms to transmit unacknowledged mode data PDUs several times.

  [Note: Whether quick repeat function is performed by layer 3 or by RLC sublayer is FFS..]

8 Ciphering

8.1 Location of ciphering function in the UTRAN protocol architecture

The ciphering function is performed either in the RLC sub-layer or in the MAC sub-layer, according to the following rules :

- If a logical channel is expected to be supported on common transport channel and has to be ciphered, it can not use the transparent mode of RLC (it should use the UM RLC mode instead).
- If a logical channel is using a non-transparent RLC mode (AM or UM), ciphering is performed in the RLC sub-layer.
- If a logical channel is using the transparent RLC mode, ciphering is performed in the MAC sub-layer (MAC-d entity).

According to this model, ciphering is always performed in the SRNC, and the context needed for ciphering (Kc, HFN, etc.) is only known in SRNC.

8.2 Ciphering algorithm

8.2.1 Overview

When ciphering is performed in the RLC sub-layer, it performs the encryption/decryption of the data part of an RLC PDU, based on XOR combining with a mask obtained as an output of the ciphering algorithm.

When ciphering is performed in the MAC sub-layer, it performs the encryption/decryption of a MAC SDU (RLC PDU), based on XOR operation with a mask obtained as an output of the ciphering algorithm.

The generic algorithm and its parameters are described in the following figure. It will be specified by SA3, and requirements are described in TS33.105.

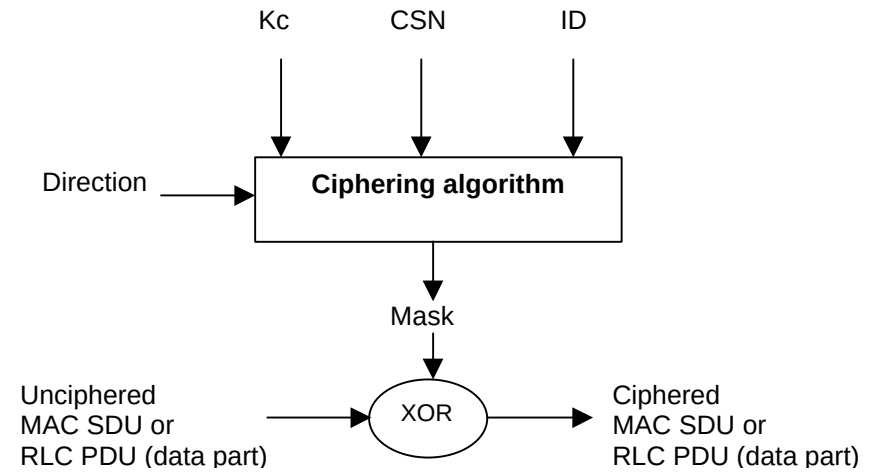

Figure 1 : Ciphering algorithm and parameters

8.2.2 Ciphering algorithms parameters

8.2.2.1 Ciphering sequence number

The ciphering sequence number (CSN) shall be at least 32 bits long. It is composed of a 'long' sequence number called Hyper Frame Number HFN, and a 'short' sequence number, which depends on the ciphering mode, as described below. There is one ciphering sequence per logical channel using AM or UM mode plus one for all logical channels using the transparent mode (and mapped onto DCH).

The Hyper Frame Number (HFN) is initialised by the UE and signalled to the SRNC before ciphering is started. It is used as initial value for each ciphering sequence, and it is then incremented independently in each ciphering sequence, at each cycle of the 'short' sequence number. When a new RAB / logical channel is created during a RRC connection, the highest HFN value currently in use is incremented, and used as initial value for the ciphering sequence of this new logical channel. The highest HFN value used during a RRC connection (by any ciphering sequence) is stored in the USIM, and the UE initialises the new HFN for the next session with a higher number than the stored one. If no HFN value is available in USIM, the UE randomly selects a HFN value.

Depending on the requirements (e.g. how many successive RRC Connections can use the same ciphering key), it may be sufficient to use only the most significant bits of HFN in the re-initialization (and set LSBs implicitly to zero). This may be necessary at least if the HFN value needs to be included in the RRC Connection Request message.

The 'short' sequence number is :

- For RLC TM on DCH, the CFN of the UEFN is used and is independently maintained in UE MAC and SRNC MAC-d. The ciphering sequence number is identical to the UEFN.
- For RLC UM and AM modes, the RLC sequence number is used, and is directly available in each RLC PDU at the receiver side (it is not ciphered). The HFN is incremented at each RLC SN cycle.

The figure below presents some examples of the different ciphering sequence numbers, assuming various sizes for the 'short' sequence numbers. This proposal permits to exchange a unique HFN and also to use a unique CSN size, which should permit to reduce the implementation complexity of the ciphering function. In this example, the HFN is 25 bits long, and only the 20 MSB are used for the CSN of the RLC AM mode.

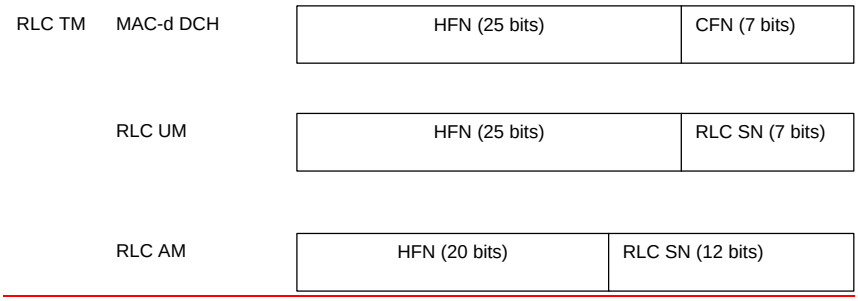

Figure 2 : Example of ciphering sequence number for all possible configurations

8.2.2.2 Ciphering key K_c

K_c is exchanged between the UE and SRNC during the authentication phase. The selection of K_c when a UE is connected with multiple CN is FFS.

8.2.2.3 ID

This parameter indicates the logical channel identity, which shall be unique within a RRC connection. It is used as input parameter of the ciphering algorithm to ensure that the same ciphering mask is not applied to two or more parallel logical channels having the same Kc and same CSN. Each logical channel is ciphered independently.

8.2.2.4 Direction

This parameter indicates the transmission direction (uplink/downlink).

~~[Note: This section describes preliminary assumptions on ciphering and a proposed solution to the frame number initialization problem. This text may be moved to other parts of this document or to other specifications (TS 25.321, TS 25.322) after decisions have been taken. The proposed scheme is not approved yet by WG2.]~~

~~Presently two solutions for the ciphering execution exist. Ciphering is performed either *(a)* always on MAC layer or *(b)* depending on RLC type (transparent vs. non-transparent) either on MAC or on RLC layer. In addition, integrity control~~

on Common Control Channels is considered. (Introduction of a separate ciphering sublayer may be considered in case there are severe problems with these assumptions).

Any ciphering algorithm requires a sequence number as an input parameter. As sequence number, a UE frame number (UE FN) as defined in TS 25.401 [2] may be used when ciphering is performed in MAC layer. The UE FN is composed at least of a Connection Frame Number (CFN). The sequence number is composed at least of the RLC PDU sequence number when ciphering is performed in RLC. To meet the requirements for the length of frame number used for ciphering, a Hyper Frame Number (HFN), incremented at every completed cycle of the CFN, is added to the UE FN. With this solution, the length of the frame number broadcast in BCCH need not be as long as the length of frame number used for ciphering (which should be at least 32 bits).

A problem with regard to security risk, common for all proposed ciphering methods, is how to initialize the HFN. In case of MAC ciphering, only one HFN is needed. In case of MAC and RLC ciphering, at least two HFNs (or equivalent counters) are needed.

Following requirements exists:

- Initialise the HFN before the ciphering is activated.
- Avoid to reuse the same SN value in input of the ciphering algorithm twice or more in a "short" time (especially with the same ciphering key, Kc). This reduces the security of the system.

A problem exists when a UE uses the same Kc (ciphering key) in two subsequent RRC connections. If in both connections the HFN is initialised to zero, the same inputs to the ciphering algorithm (FN and Kc) are used twice and the same ciphering mask may be reused in a relatively short period of time. This may occur also with a random initialisation of HFN. This is not 'secure' and should be avoided.

One possible solution to this is that when the RRC connection is released, the terminal (SIM card) stores the last HFN(s) used.

At a new RRC connection setup or at RRC connection re-establishment, UE initialises the HFN(s) to a value higher than the last used HFN(s) value, and transmits this/these to the SRNC, either in the RRC Connection Request message or in the first message after the RRC connection is established.

Depending on the requirements (e.g. how many successive RRC Connections can use the same ciphering key), it may be sufficient to use only the most significant bits of HFN in the re-initialization (and set LSBs implicitly to zero). This may be necessary at least if the HFN value needs to be included in the RRC Connection Request message.