EUROPEAN TELECOMMUNICATION STANDARD

DRAFT
pr **ETS 300 392-7**
September 1995

Source: ETSI TC-RES

Reference: DE/RES-06001-7

ICS: 30.060.50

Key words: TETRA, V+D

Radio Equipment and Systems (RES); Trans-European Trunked Radio (TETRA); Voice plus Data (V+D); Part 7: Security

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

Tel.: +33 92 94 42 00 - Fax: +33 93 65 47 16

Contents

Foreword

This draft European Telecommunication Standard (ETS) has been produced by the Radio Equipment and Systems (RES) Technical Committee of the European Telecommunications Standards Institute (ETSI), and is now submitted for the Public Enquiry phase of the ETSI standards approval procedure.

This ETS is a multi-part standard and will consist of the following parts:

Part 1: "General network design".

Part 2: "Air Interface (AI)".

Part 3: "Inter-working - Basic Operation", (DE/RES-06001-3).

Part 4: "Gateways for Basic Services", (DE/RES-06001-4).

Part 5: "Terminal equipment interface", (DE/RES-06001-5).

Part 6: "Line connected stations", (DE/RES-06001-6).

Part 7: "Security".

Part 8: "Management services", (DE/RES-06001-8).

Part 9: "Performance objectives", (DE/RES-06001-9).

Part 10: "Supplementary Services (SS) Stage 1".

Part 11: "Supplementary Services (SS) Stage 2", (DE/RES-06001-11).

Part 12: "Supplementary Services (SS) Stage 3", (DE/RES-06001-12).

Part 13: "SDL Model of the Air Interface", (DE/RES-06001-13).

Part 14: "PICS Proforma", (DE/RES-06001-14).

Part 15: "Interworking - Extended Operations", (DE/RES-06001-15).

Part 16: "Gateways for Supplementary Services", (DE/RES-06001-16).

Proposed transposition dates	
Date of latest announcement of this ETS (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Blank page

1 Scope

This ETS describes the security mechanisms in the Trans-European Trunked Radio (TETRA) Voice + Data (V+D) standard. It provides mechanisms for confidentiality of control signalling and user speech and data at the air interface, authentication and key management mechanisms for the air interface, and end-to-end confidentiality mechanisms between users.

Clause 4 describes the confidentiality mechanisms using encryption on the air interface, for circuit mode speech, circuit mode data, packet data and control information.

Clause 4 also describes both encryption mechanisms and mobility procedures. It also details the protocol concerning control of encryption at the air interface.

Clause 5 describes the authentication and key management mechanisms for the TETRA air interface. The following two authentication services have been specified for the air-interface in ETR 086-3 [3], based on a threat analysis:

- authentication of a user by the TETRA infrastructure;

- authentication of the TETRA infrastructure by a user.

Clause 6 describes the end-to-end confidentiality for V + D. End-to-end confidentiality can be established between two users or a group of users.

In clause 6 the logical part of the interface to the encryption mechanism is described. Electrical and physical aspects of this interface are not described, nor are the key management and encryption algorithms for end-to-end confidentiality described.

The encryption mechanism described in this ETS allows several options for protection against replay.

2 Normative references

This ETS incorporates by dated and undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references the latest edition of the publication referred to applies.

[1] prETS 300 392-1: "Radio Equipment and Systems (RES); Trans-European Trunked Radio (TETRA); Voice plus Data (V+D); Part 1: General network design".

[2] prETS 300 392-2: "Radio Equipment and Systems (RES); Trans-European Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI)".

[3] ETR 086-3: "Radio Equipment and Systems (RES); Trans European Trunked Radio (TETRA) systems; Technical requirements specification; Part 3: Security aspects".

[4] ISO 7498-2: "Information processing systems - Open Systems Interconnection - Basic reference model - Part 2: Security Architecture".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETS, the following definitions apply:

Authentication Code (AC): A (short) key to be entered by the user into the terminal.

Authentication Key (K): The primary secret, the knowledge of which has to be demonstrated for authentication. On the infrastructure side, it is stored in a secure place of the home network. In the terminal it is generated in one of three ways: 1) the authentication key may be generated from an authentication code AC that is manually entered by the user; 2) the authentication key may be generated from a user authentication key UAK stored in a module (detachable or not); 3) the authentication key may be generated from both the UAK stored in a module and the PIN entered by the user.

Cipher Key (CK): A value that is used to determine the transformation of plain text to cipher text in a cryptographic algorithm.

cipher text: The data produced through the use of encipherment. The semantic content of the resulting data is not available (ISO 7498-2) [4].

Common Cipher Key (CCK): Generated by the infrastructure to protect group calls. There is one CCK for each location area.

decipherment: The reversal of a corresponding reversible encipherment (ISO 7498-2) [4].

Derived Cipher Key (DCK): Calculated from two parts, DCK1 and DCK2, (in the case of unilateral authentication either DCK1 or DCK2 is set to zero). DCK1 is generated during authentication of the user by the TETRA infrastructure. DCK2 is generated during authentication of the TETRA infrastructure by the user.

derived key: A sequence of symbols that controls the KSG inside the end-to-end encryption unit and that is derived from the CK.

encipherment: The cryptographic transformation of data to produce cipher text (ISO 7498-2) [4].

end-to-end encryption: The encryption within or at the source end system, with the corresponding decryption occurring only within or at the destination end system.

flywheel: A mechanism to keep the KSG in the receiving terminal synchronised with the KSG in the transmitting terminal in case synchronisation data is not received correctly.

Initialisation Value (IV): A sequence of symbols that initialises the KSG inside the encryption unit.

Key Number (KN): Distributed with the CCK. It serves the identification of the active key and the protection against replay of old keys.

key stream: A pseudo random stream of symbols that is generated by a KSG for encipherment and decipherment.

Key Stream Generator (KSG): A cryptographic algorithm which produces a stream of binary digits which can be used for encipherment and decipherment. The initial state of the KSG is determined by the initialisation value.

Key Stream Segment (KSS): A key stream of arbitrary length.

Manipulation Flag (MF): Used to indicate that the CCK has been incorrectly recovered.

Personal Identification Number (PIN): Entered by the user into the terminal and used to generate the authentication Key (K)together with the User Authentication Key (UAK).

plain text: The unencrypted source data. The semantic content is available.

proprietary algorithm: An algorithm which is the intellectual property of a legal entity.

Random challenge (RAND1, RAND2): A random value generated by the infrastructure to authenticate a user or in a terminal to authenticate the infrastructure, respectively.

Random Seed (RS): A random value used to derive a session authentication key from the authentication key.

Response (RES1, RES2): A value calculated in the terminal from RAND1 and the KS to prove the authenticity of a user to the infrastructure or by the infrastructure from RAND2 and the KS' to prove its authenticity to a user, respectively.

Sealed Common Cipher Key (SCCK): A common CK, cryptographically sealed with a particular user's derived CK. In this form the keys are distributed over the air interface.

Session Authentication Key (KS, KS'): Generated from the authentication key and a random seed for the authentication of a user. It has a more limited lifetime than the authentication key and can be stored in less secure places and forwarded to visited networks.

spoofer: An entity attempting to obtain service from or interfere with the operation of the system by impersonation of an authorised system user or system component.

Static Cipher Key (SCK): A predetermined CK used if no (successful) authentication has taken place.

Stream Cipher Algorithm (SCA): A cryptographic system for which plain text and cipher text are processed in a continuous stream.

synchronisation value: A sequence of symbols that is transmitted to the receiving terminal to synchronise the KSG in the receiving terminal with the KSG in the transmitting terminal.

synchronous stream cipher: An encryption method in which a cipher text symbol completely represents the corresponding plain text symbol. The encryption is based on a key stream that is independent of the cipher text. In order to synchronise the KSGs in the transmitting and the receiving terminal synchronisation data is transmitted separately.

TETRA algorithm: The KSG inside the encryption unit.

time stamp: Is a sequence of symbols that represents the time of day in a year.

User Authentication Key (UAK): Stored in a (possibly detachable) module within the terminal and used to derive the authentication key (with or without a PIN as an additional parameter).

3.2 Abbreviations

For the purposes of this ETS the following abbreviations apply. Where SDL equivalents are used, these are shown in parentheses, thus: (SDL_equivalent).

AC	Authentication Code
AI	Air Interface
AN	Algorithm Number (algorithm_number)
AS	Alias Stream
BS	Base Station
Call-ID	Call Identifier (call_id)
CCK	Common Cipher Key
CID	Cipher Identifier (cipher_id)
CK	Cipher Key (cipher_key)
C-PLANE	Control Plane
CR	Cipher Report (cipher_report)
CT	Cipher Text
DCK	Derived Cipher Key
DCK1	Part 1 of the DRC
DCK2	Part 2 of the DRC
DK	Derived Key
ES	Encryption Switch (encryption_switch)
F	Function
HSC	Half-Slot Condition (half_slot_condition)
HSI	Half-Slot Importance (importance)
HSN	Half-Slot Number (half_slot_number)
HSS	Half-Slot Stolen (stolen_indication)
HSSE	Half-Slot Stolen by Encryption unit
ITSI	Individual TETRA Subscriber Identity
IV	Initialisation Value (init_value)
K	authentication Key
KC	Key Command (key_command)
KN	Key Number (key_number)
KS	Session authentication Key
KSS	Key Stream Segment
LLC	Logical Link Control
MAC	Medium Access Control
MF	Manipulation Flag
MLE	Mobile Link Entity
MM	Mobility Management
MS	Mobile Station
PDU	Protocol Data Unit
PIN	Personal Identification Number
PT	Plain Text
RAND1	Random challenge 1
RAND2	Random challenge 2
RCS	Reception Status
RES1	Response 1
RES2	Response 2
RO	Replay Offset (replay_offset)
ROS	Replay Offset Select (replay_offset_select)
RS	Random Seed
SAP	Service Access Point
SCK	Static Cipher Key
SCCK	Sealed Common Cipher Key
SDU	Service Data Unit
SF	Synchronisation Frame
SHSI	Stolen Half-Slot Identifier

SR	Slot Rate (slot_rate)
SS	Synchronisation Status
SV	Synchronisation Value
SwMI	Switching and Management Infrastructure
TA	TETRA Algorithm
TCH	Traffic Channel type (TCH_type)
TRS	Transmit Status-
UAK	User Authentication Key
UD	User Device (u_device)
U-PLANE	User Plane
XRES1	Expected response 1
XRES2	Expected response 2

4 Air Interface (AI) encryption

4.1 Encryption mechanism

4.1.1 General principles

AI encryption shall provide confidentiality on the radio link between Mobile Station (MS) and Base Station (BS).

AI operates by combining the output of a Key Stream Generator (KSG) with the contents of messages to be transmitted across the air interface. Both control and traffic (speech or data) information can be encrypted. The encryption process takes place in the upper Medium Access Control (MAC) layer of the TETRA protocol stack.

The headers for each message (i.e. each MAC Packet Data Unit (PDU)) shall not be encrypted. This shall enable an MS to determine which messages are destined for it, the PDU type, the encryption state of the PDU's, and so determine which of these messages require decryption for further action. It shall also enable a BS to identify the individual MS or group involved in communication, and so select the appropriate key.

An alias mechanism is provided which enables addresses contained in MAC headers, and hence the identities of the MS's involved in communication, to be concealed from eavesdropping.

Channel information that is contained in the headers of MAC PDUs may be encrypted also using the encryption mechanism. This is further described in subclause 4.1.3.3.

Each MS should hold a secret key which is used to determine the Cipher Key (CK) used by the KSG. Additionally, a common key should be used for group addressed signalling. The BS shall have knowledge of all CKs in use by all individual MSs and all groups that are registered at that BS, or in the location area of which the BS is a part.

The KSG shall form an integral part of an MS or BS. It shall not be provided as a defined interface to a separate unit.

Air Interface encryption shall be a separate function to the end-to-end encryption service described in clause 6. Information that has been encrypted (already) by the end-to-end service can be encrypted again by the air interface encryption function. Where TETRA provides for clear or encrypted call services in ETS 300 392-1 [1], these shall be independent of air interface encryption; thus a call service invoked without (end-to-end) encryption may still be encrypted at the air interface.

4.1.1.1 Key Stream Generator (KSG)

Encryption shall be realised using an encryption algorithm, a so-called KSG.

The KSG shall have two inputs, an Initial Value (IV) and a CK. These parameters are specified in subclause 4.1.2. The KSG produces one output:

- a sequence of key stream bits of arbitrary length, referred to as a Key Stream Segment (KSS).

A key stream segment (KSS) of length K shall be produced to encrypt every slot. The bits of KSS are labelled KSS(0), .KSS(K-1), where KSS(0) is the first bit output from the generator. The bits in the KSS shall be used to encrypt or decrypt the data of the control or traffic field. The maximum value of (K-1) shall be 432, which enables encryption of a TCH/7,2 unprotected traffic channel.

4.1.1.2 Encryption mechanism

The key stream bits shall be modulo 2 added (XORed) with plain text bits in data, speech and control channels to obtain encrypted cipher text bits, with the exception of the MAC header bits and fill bits. KSS(0) shall be XORed with the first transmitted bit of the first MAC SDU, and so on. There shall be one exception to this procedure which occurs when the MAC header includes a channel numbering field. This is described in subclause 4.1.3.3.

If the information in a slot has fewer bits than the length of KSS produced, the last unused bits of KSS shall be discarded. For example, if there are M information bits, KSS(0) to KSS(M-1) shall be utilised, KSS(M) to KSS(K-1) shall be discarded.

The MAC may perform PDU association, where more than one PDU may be transmitted within one slot. These PDUs may be addressed to different identities. The MAC headers themselves may be of varying lengths. To allow for this, the KSS shall be restarted at the commencement of each SDU; the KSS that encrypts each SDU should be different provided that the SDUs within one slot are addressed to different identities, because the KSSs should be produced with different keys.

The MAC headers shall not be encrypted. Also, fill bits used at the end of an SDU to make the SDU up to a complete number of bytes shall not be encrypted. Figure 1 illustrates this process.

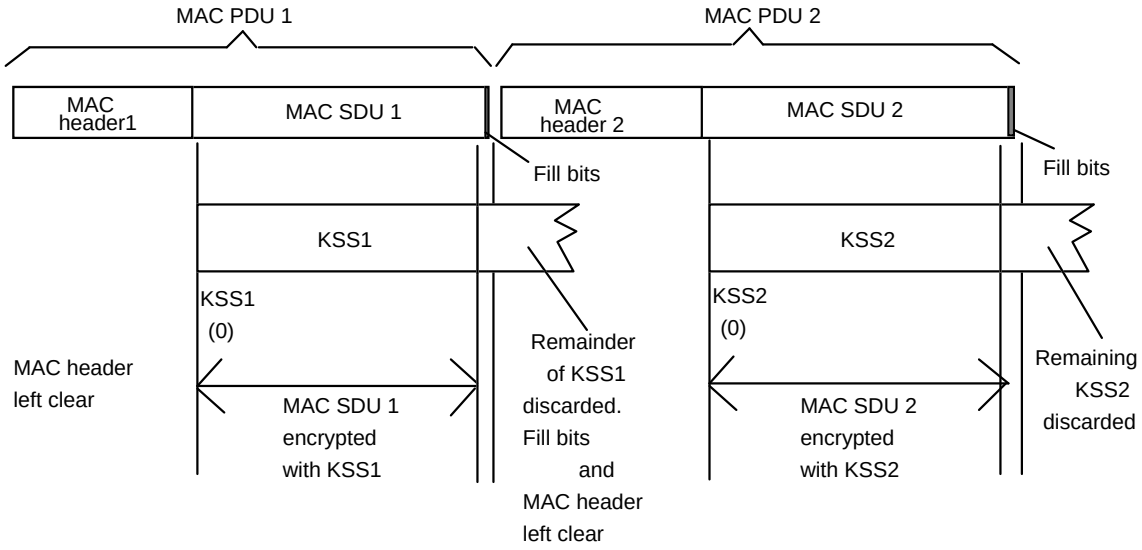


Figure 1: Allocation of KSS to encrypt MAC PDUs

Where a transmitted slot is divided into subslots or half slots, the same principle shall apply, with KSS being restarted at the start of each SDU. Subslots and half slots are described in ETS 300 392-2 [2].

To avoid replay of key stream, sending more than one SDU addressed to the same identity within one slot should be avoided.

On a traffic channel KSS shall only be initialised at the start of a slot, even if half of the slot is stolen.

4.1.1.3 KSG numbering and selection

There is one TETRA standard KSG. Proprietary KSGs may also be supported. Signalling permits different KSGs to be identified. Migration is only possible if there is agreement between operators on the KSG used.

The TETRA standard algorithm KSG shall only be available on a restricted basis from ETSI.

In the normal case, only one KSG should be supported within a system; however it is possible that more than one KSG may be implemented. The implementation of a system with more than one KSG is outside the scope of this ETS.

4.1.2 Interface parameters

4.1.2.1 Initial Value (IV)

The IV shall be used to initialise the KSG at the start of every slot. The IV shall be a value 29 bits long represented as IV(0)....IV(28) based on the frame numbering system, where IV(0) shall be the least significant bit and IV(28) the most significant bit of IV.

The composition of the IV shall be as follows:

- the first two bits IV(0) and IV(1) shall correspond to the slot number, and shall take values from 0 to 3, where value 0 corresponds to slot 1, and value 3 corresponds to slot 4. IV(0) shall be the least significant bit of the slot number;

- the next five bits IV(2) to IV(6) shall correspond to the frame number, and shall take values from 1 (00001 binary) to 18 (10010 binary). IV(2) shall correspond to the least significant bit of the frame number;

- the next six bits IV(7) to IV(12) shall correspond to the multiframe number, and shall take values from 1 (00001 binary) to 60 (111100 binary). IV(7) shall correspond to the least significant bit of the multiframe number;

- the next 16 bits IV(13) to IV(28) shall correspond to an extension, that numbers the hyperframes. These can take all values from 00 to 65535. IV(13) shall correspond to the least significant bit of the hyperframe numbering extension.

4.1.2.2 Cipher Keys (CKs)

Figure 2 shows the ciphering process. A CK shall be used in conjunction with a KSG to generate a key stream for encryption and decryption of information at the MAC layer. It can be considered a binary vector of 80 bits, labelled CK(0) ... CK(79). The CK used for encryption and decryption of the uplink may be different from the CK used for encryption and decryption of the downlink, as described in subclause 4.1.2.3.

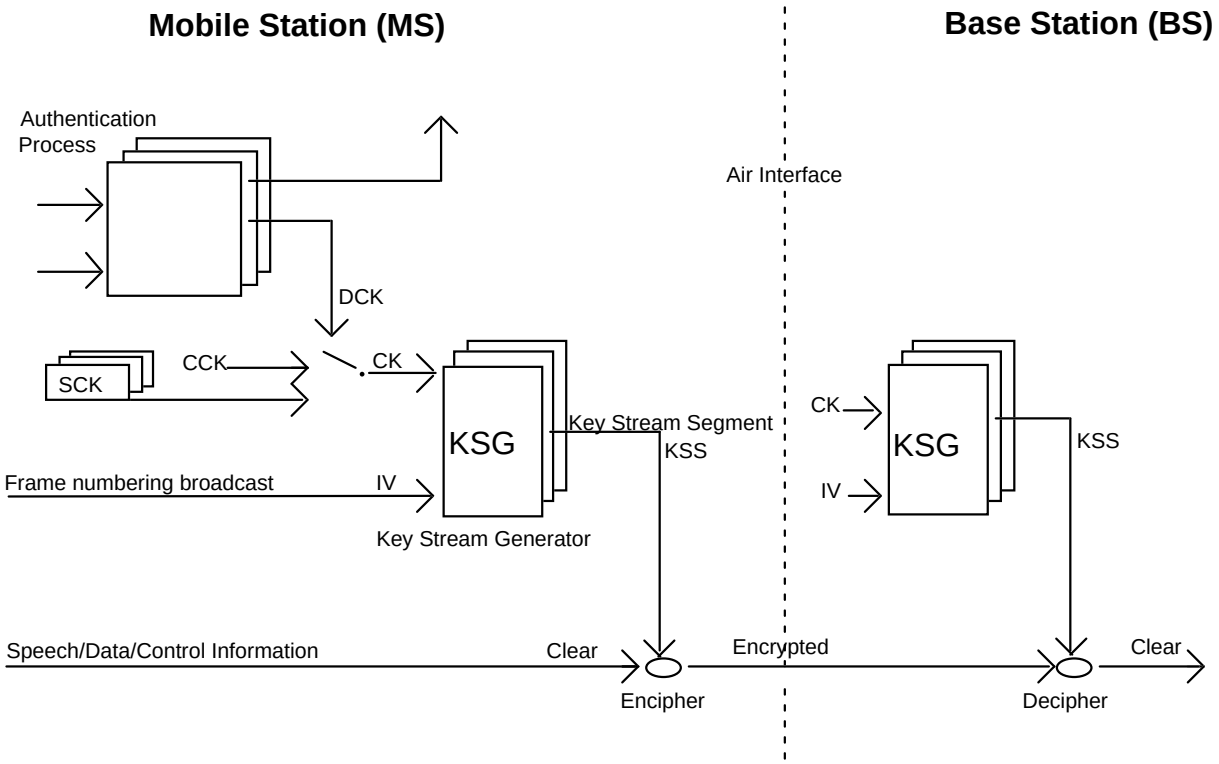


Figure 2: Speech and control information encryption

Three types of CKs are defined:

- the Static CK (SCK);
- the Derived CK (DCK); and
- the Common CK (CCK).

These are described in subclause 4.1.2.3.

4.1.2.3 Allocation of CKs

The Static CK (SCK) may be uniquely allocated to a single ITSI, or may be allocated to multiple ITSIs. It can be chosen by the system manager and manually entered in both MS and SwMI. It may have an indefinite lifetime. The allocation of an SCK shall be carried out in the home network of the MS. The choice and distribution of the SCK is outside the scope of this ETS.

The Derived CK (DCK) shall be generated from the authentication of the MS by a process using a secret key. A new DCK is established at each authentication, as described in clause 5.

Only one DCK shall be associated with a single ITSI at any time.

A Common CK (CCK) shall be associated with a Location Area (LA). There may be more than one version of a CCK associated with each LA to allow for future use. The current CCK shall be sent to the MS after registration and authentication, encrypted using the DCK established at authentication, as described in clause 5. Future versions may also be downloaded in this manner.

One CCK shall be used for all group addressed signalling to all MSs within a location area. Due to this widespread use, it is necessary to incorporate a mechanism to periodically change the CCK in use. A new CCK shall be individually sent to each MS encrypted using its DCK. The CCK in use shall be signalled in each downlink message by the BS. As the process of sending a new CCK to each MS within a location area individually may take some time to achieve, an MS shall be able to store more than one CCK at a time.

4.1.2.4 Usage of CKs

The header of MAC PDUs transmitted over the air interface shall contain indications of the key in use.

The SCK may be used without the need for registration and authentication. It can be used for all transmissions, uplink or downlink. It should be used in preference to using no encryption on systems where authentication is not implemented.

Use of a DCK on the air interface should be optional. Once a DCK has been established for an MS, the DCK may be used for all uplink transmissions from that MS, and for all downlink transmissions intended for that MS alone.

One CCK shall be used for all downlink transmissions to all groups of MSs within one location area. Downlink transmissions shall contain bits which signal the CCK in use, this shall permit an MS to store two CCKs at once whilst a change in CCK is occurring.

Once a type of key, SCK or DCK, and a set of ciphering parameters has been established, a change to key type or ciphering parameters shall only be made on command from the SwMI. Mobility Management (MM) in the MS shall only change the ciphering parameters of the MS in response to such a request. This procedure is described in subclause 4.3.2.1.

Initial registration and authentication shall always be carried out without any encryption applied. This shall be the case even if the SCK is the only CK in use on a system. It shall also be the case even if either of the registration or authentication procedures are not implemented. Re-registrations may be carried out with encryption applied, unless re-registration is forced due to a mis-match in encryption parameters.

Should circumstances ever arise in which the MS is permitted to place or take part in a call made prior to registration, the call may either be made without encryption, or it may be encrypted using the SCK. This selection shall be prearranged between MS and SwMI.

An MS may store more than one CCK to aid mobility between location areas. When an MS roams to a new location area, it may retain the CCKs that were used in the previous location area to permit roaming back without the need to obtain the keys again.

If an MS and SwMI load different keys from each other, the receiving party will decode messages incorrectly. This will cause erroneous operation. The result of this, and any corrective action put in place to prevent errors, for example attempting a re-authentication to establish new keys, is outside the scope of the ETS. The use of DCK and CCK is illustrated in figure 3.

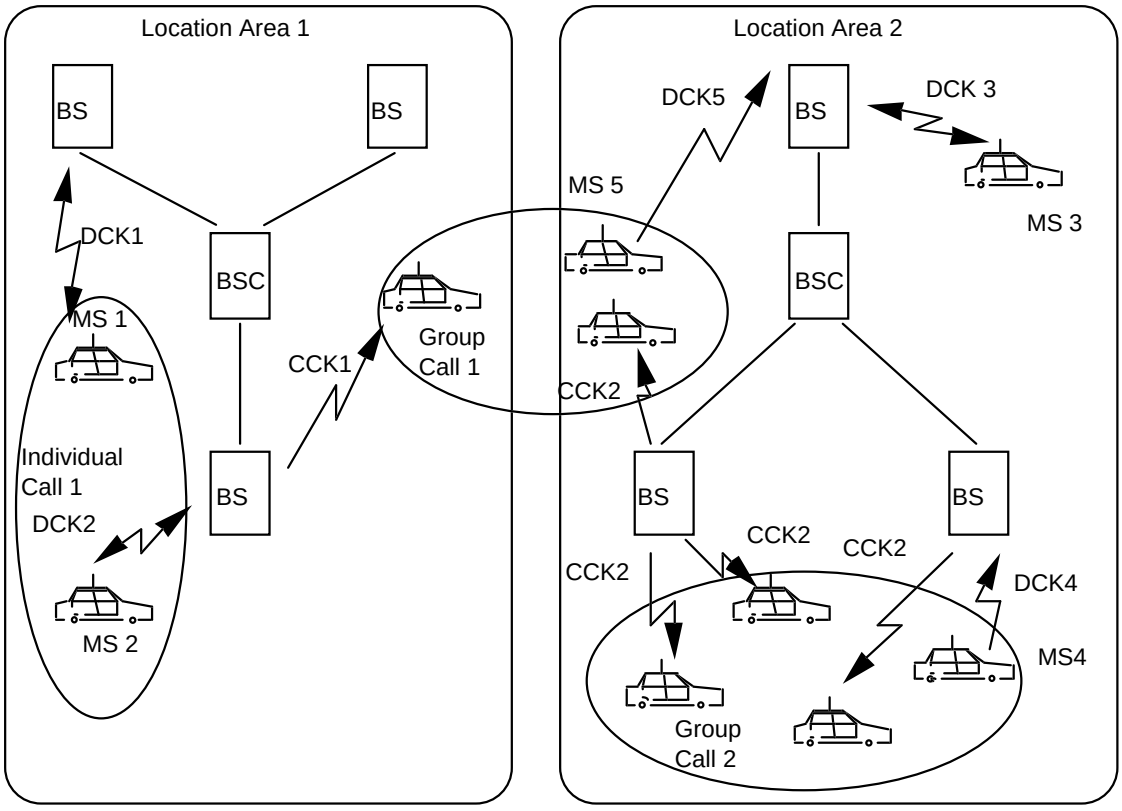


Figure 3: Illustration of Derived and Common CK usage

4.1.2.5 Identification of CKs

Two bits of the headers of downlink MAC PDUs shall be reserved for air interface encryption management, and shall be used to indicate whether encryption is in use, and if so, which is the current key in use.

The values of these two bits shall be assigned on the downlink for group addressed signalling as follows:

- 00 = No encryption;
- 01 = Encrypted with SCK;
- 10 = Encrypted with CCK key 1;
- 11 = Encrypted with CCK key 2.

The values of these bits shall be assigned on the downlink for individually addressed signalling as follows:

- 00 = No encryption;
- 01 = Encrypted with SCK;
- 10 = Encrypted with DCK;
- 11 = Reserved.

This mechanism shall not be used to select between multiple CCKs in use at the same time within a LA. It shall only be used as a means of selecting a new key within a LA once this new key has been communicated to all registered MSs.

To prevent attacking by replaying a previous key, the CCK shall be identified by a longer Key Number (KN) which shall be sent to an MS together with the CCK. The least significant bit of the KN shall be identical to the least significant bit of the CK identifier in the MAC header when the most significant bit of the header is set to indicate CCK in use.

Furthermore, one bit of uplink MAC PDU headers shall also be reserved for air interface encryption. This shall indicate whether the contents of the PDU are encrypted or not.

This bit shall take one of the following values:

- 0 Encryption off;
- 1 Encryption on.

If it is desired to change the DCK in use by an MS, this should be achieved by the normal authentication process; and as both BS and MS are involved in the process and have knowledge that it has occurred, it shall not be necessary to include a key identifier in the uplink header.

The indication bits shall also indicate the state of the channel number encryption system as described in subclause 4.1.3.3, and the identity alias system as described in subclause 4.1.6.

4.1.3 Data to be encrypted

4.1.3.1 Encryption of MAC PDUs

This subclause describes the method of applying the air interface encryption to PDUs in the upper MAC layer.

An SDU is passed down to the MAC from the LLC, and the MAC adds a header to this to form a MAC PDU. To enable the identity to be distinguished at the receiving side of the air interface, the encryption process shall not be applied to the MAC header, but applied to the contents of the MAC SDU only. This is described in subclause 4.1.1.2 and shown in figure 1.

The content of the MAC SDU is transparent to the MAC, it may be already encrypted end-to-end in circuit mode.

Where communication originates in the MAC layer and is destined for the peer MAC across the air interface, the MAC SDU consists only of a header. This shall not be encrypted.

Traffic channels do not incorporate a separate MAC header in the same way as control channels, except where half slots are stolen for signalling purposes. Instead, the entire channel is used for traffic data. Therefore on a traffic channel, the SDU that is encrypted is the entire contents of the transmitted slot.

Where a slot or part of a slot is stolen, the MAC header is determined by the use of the slot. If it is used for U-plane data, a three bit MAC header is included to define the use of the stolen slot. If it is used for C-plane data, the header is longer, but the first two bits are common with the U-plane implementation to define the use of the slot. In either case, the header shall not be encrypted.

4.1.3.2 Control channel requirements

Certain control messages shall not be encrypted, as they shall be used by MSs prior to authentication and so prior to establishment of a DCK.

The messages that shall not be encrypted when transmitted from a BS are:

- all messages sent to the MAC via the TMB-SAP;

- all messages sent to the MAC via the TMA-SAP that are addressed to all MSs. These are messages where the destination SSI is set to all “1”s.

All remaining messages originating from higher layers shall be encrypted if a channel has been switched to encrypted operation.

When transmitted from an MS, all messages originating from higher layers shall be encrypted following authentication, as described in subclause 4.1.2.4.

However, the MAC PDU headers shall not be encrypted on any messages.

4.1.3.3 Encryption of channel numbering information

Channel numbering information is sent in the downlink MAC-RESOURCE PDU (used for channel allocation, cell change, control channel change etc.). There are several fields, of which some shall be encrypted as follows in table 1.

Table 1: Additional ciphering elements

Field	Size	Type	Encryption state
Channel allocation type	2 bits	Mandatory	Encrypted
Time slot	2 bits	Mandatory	Encrypted
Up/down link	2 bits	Mandatory	Encrypted
CLCH permission	1 bit	Mandatory	Encrypted
Cell change flag	1 bit	Mandatory	Encrypted
Carrier number	12 bits	Mandatory	Encrypted
Extended carrier number flag	1 bit	Mandatory	Not encrypted
(Frequency band	4 bits)	Optional	Not encrypted
(Channel offset	2 bits)	Optional	Not encrypted
(Duplex spacing	3 bits)	Optional	Not encrypted
(Reverse operation	1 bit)	Optional	Not encrypted
Monitoring pattern	2 / 4 bits	Mandatory	Not encrypted

The extended carrier flag, optional fields and monitoring pattern field shall not be encrypted, to prevent difficulties calculating the length of a PDU by an MS not possessing the correct CK.

The encryption process shall be accomplished in the same manner as is used to encrypt MAC SDUs, i.e. the modulo 2 addition of a key stream, where the key stream shall be generated as a function of frame numbering and CK relevant to the addressed party or parties. Therefore, if this information is sent to an individual MS, it shall be encrypted using the DCK relevant to that MS. If it is sent to a group, it shall be encrypted using the CCK for that location area. If it is sent to all MS's registered on that site, it should be also encrypted using the CCK for the LA.

The KSG shall be initialised as described in subclause 4.1.2.1 using the frame and slot numbering system.

The PDU header may have an SDU attached to it; which would also be encrypted using the same CK in conjunction with the frame numbering information. To prevent a key stream repeat, the first 20 bits of the key stream shall be used to encrypt channel numbering information, and the remaining bits shall be used to encrypt the SDU.

Thus where channel numbering information is present, identified by a bit in the MAC header, the first 20 bits of key stream (KSS[0] to KSS[19]) shall be allocated to the channel numbering information. The remainder of the key stream, KSS[20] onwards, shall be used to encrypt the SDU. Where channel numbering information is not present, KSS[0] onwards shall be used to encrypt the SDU.

The channel numbering information in the MAC header shall have the same encryption state as the SDU, with encryption state indicated by bits in the header as described in subclause 4.1.2.5.

To avoid a key stream repeat, the encrypted PDU should not be sent in the same time slot as another PDU encrypted with the same key.

4.1.3.4 Traffic channel requirements

Traffic channels may be transporting speech or data. The information shall be encrypted prior to channel encoding.

The state of encryption on a traffic channel shall follow the state of encryption used on the control channel to set the call up. Therefore, if the call was set up using encrypted signalling on the control channel, the traffic channel shall also be encrypted.

4.1.4 Initialisation and synchronisation of encryption process

4.1.4.1 Initialisation

Prior to registration, an MS shall operate only in clear mode; hence no encryption shall be established in its MAC layer. The only exception to this may occur should the MS be permitted to take part in a call prior to registration, as described in subclause 4.2.4. The MS shall receive information that is broadcast by the SwMI in clear form. Information transmitted by the SwMI other than broadcasted system information may be encrypted, and so the MS may not be able to make use of such information. Once cell selection is completed, the MS may register to establish encryption parameters. Once this is complete, the MS shall activate the encryption process in its MAC layer if the SwMI so demands; after that all control messages between that MS and SwMI can be encrypted as described in subclause 4.1.3.2.

The KSG shall be re-initialised with frame numbering information at the start of every slot which is to be encrypted.

4.1.4.2 Synchronisation

The MS and the SwMI shall rely on synchronised frame numbering to maintain synchronisation of their key stream segments. The IV loaded into the KSG shall be derived from this, as described in subclause 4.1.2.1.

The method for assigning key stream bits to the bits of the MAC PDUs is described in subclause 4.1.1.2.

4.1.5 Encryption mode control

Encryption of control and speech/data channels shall be switched on and off only by the SwMI; i.e. an MS shall not originate a change of state of encryption mode.

Encryption mode control is achieved by an exchange of MM PDUs. The PDU exchange shall allow switching both from clear to encrypted mode and the reverse.

An MS may indicate its current encryption state to its user.

4.1.6 Identity alias system

The identity alias system shall provide a means of protection of identities transmitted over the air interface. It operates in addition to the Alias Short Subscriber Identity (ASSI) mechanism, and the two mechanisms may be used together.

NOTE: In standard TETRA addressing no alias addresses are associated with a group address. The AS mechanism provides such an alias within a location area for all address types.

A 24 bit Alias Stream (AS) shall be randomly chosen by the SwMI and transmitted to each MS in conjunction with the CCK following authentication. One AS shall always be associated with one CCK. The AS shall be encrypted using the DCK of an MS to send it to that MS across the air interface.

The MS shall decrypt the AS, and shall modulo 2 add (XOR) it with its individual identity (or alias identity) and with all group identities that it uses. By doing so, the MS shall create an alias identity table that it shall now use for sending and receiving signalling across the air interface.

Whenever signalling is used, the alias identity shall be sent instead of the true identity.

The AS shall always be transmitted in conjunction with a CCK, and used for the lifetime of that CCK. This enables the SwMI to change the AS whenever the CCK is changed; alternatively, the same AS may be sent with more than one CCK if a longer AS lifetime is desired.

The SwMI may either use a common AS in all location areas, or it may choose to use a different AS in each location area. The MS should always store an AS in conjunction with the CCK to permit roaming.

If an MS and SwMI support forward registration, the SwMI should send the AS for the new area to the MS together with the CCK for the new area to enable the MS to change cell without loss of ciphering or of identity aliassing.

The bits incorporated in the MAC header to indicate encryption control shall also indicate application of AS. Thus, if the bits are set to “0”, encryption off, the AS shall not be used in that PDU, and the true identity shall be transmitted. This enables a clear registration to be carried out with the MS's true identity visible.

To prevent error, an MS shall register without encryption in a new location area unless it already knows the CCK for that area and the associated AS.

An AS shall be associated with each SCK. Only one SCK may be in use on a system at a time.

A SwMI may transmit a large amount of information to the broadcast address (all “1”s); which could enable an adversary to identify such messages, and so derive the AS from the aliassed address. Thus, the broadcast (all “1”s) address shall not be encrypted; and the AS shall be chosen such that none of the registered users of the system can give an output of all “1”s when XORed with the AS.

4.2 Mobility procedures

4.2.1 General requirements

The cell selection procedures are defined in ETS 300 392-2 [2], the different types of communication recovery are:

- announced cell re-selection type 1, type 2 and type 3:
    - type 1: the new traffic channel is known in advance before making the decision of changing to the new cell. This is seamless. (V+D only);
    - type 2: the MS knows the new cell but not the channel allocated. Network information cannot be given by the present cell, it shall be given by the new cell;
    - type 3: the MS does not know the new cell beforehand. Services shall be interrupted for a while;
- unannounced cell reselection:
    - The MS does not inform its serving cell of the intention to change;
- undeclared cell re selection:
    - the MS does not declare the cell change to either the serving cell or the new cell.

The transfer of security information should be made entirely within the TETRA network and should not involve any unprotected transmission on the air interface, this is especially true when transferring the CK. If this protected transfer is impossible then a new CK shall be established, requiring re-authentication.

Ciphering may be interrupted and the cell change may be in clear, this is described later in this clause.

When an MS moves between LAs, it may still retain the CCK in use in the previous LA. This may permit the MS to move back without the need to obtain the CCK again.

The MS shall assume that the SwMI shall transfer its current DCK to the new cell within the network and shall use the same DCK in the new cell. If the SwMI is unable to do this, it shall require the MS to re-authenticate and establish a new DCK.

To facilitate mobility, each CCK sent to an MS shall be sent together with an identifier of the LA or cell in which that CCK is in use.

4.2.2 Mobility within a location area

Where more than one BS is present within a LA, the other BSs in the LA shall also be sent the information required to generate the DCK for each MS (i.e. random number and session authentication key) according to clause 5. Thus all BSs in the LA shall have knowledge of the DCK for every MS without the DCK having been sent across the infrastructure.

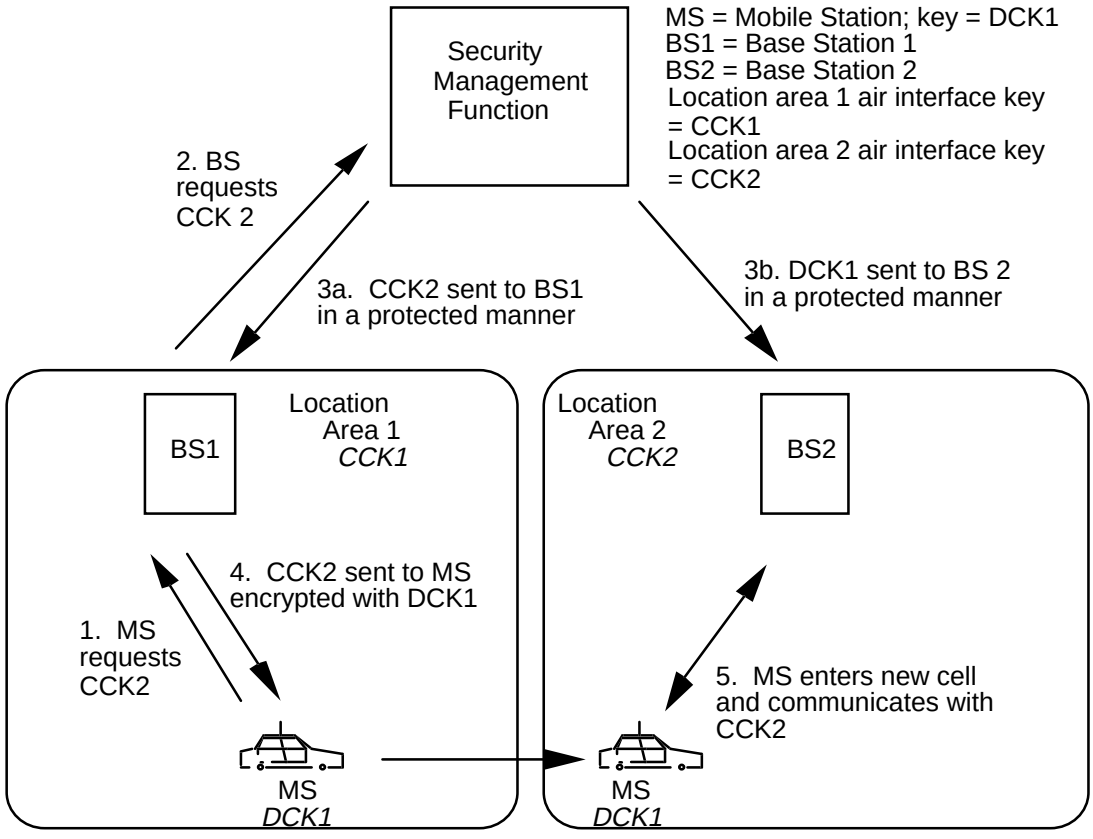
When the MS needs to move between cells, it shall obtain the CCK for the new cell. If the new cell is within the same LA, the CCK in use shall be the same; therefore the MS may be able to achieve a seamless handover without requiring re-authentication or exchange of new keys. The reselection process used may therefore not be impacted by the encryption process.

4.2.3 Mobility between location areas

To achieve mobility between areas, two methods shall be possible. Either the MS may be given the CCK in use in the new area before leaving the old area, or it may be required to register in the new area, and obtain the new CCK. If the SwMI is unable to transfer the old DCK of the MS, the MS shall also authenticate and generate a new DCK. This second method is likely to prevent a seamless handover from taking place.

If key data is to be exchanged between MS and BS, it shall be encrypted using the DCK of the MS to protect the data over the air interface. However, as the key data will have been exchanged over the internal fixed links of the SwMI, such data should be protected. NOTE: In step 3b, DCK1 needs to be sent to all base stations in new location area, unless DCK is changed by authenticaton.

Figure 4 illustrates this process.



NOTE: In step 3b, DCK1 needs to be sent to all base stations in new location area, unless DCK is changed by authenticaton.

Figure 4: Transfer of key between cells as mobile roams

The following subclauses set out the possible methods of achieving mobility between areas.

4.2.3.1 Announced reselection types 1 and 2 (new cell known)

In announced cell reselection types 1 and 2, normal operation shall be that the new cell is known before the old cell is relinquished. The MS may therefore request the CCK in use in the new cell. This may be sent by the BS in the old cell, encrypted with the DCK of the MS. At the same time, the DCK in use by the MS in the old cell may be sent to BSs in the new cell in a protected manner within the SwMI. A call in process need not then be interrupted by the requirement to authenticate and obtain new keys. This requires that the SwMI should be able to transfer keys over its internal links in a protected manner.

Once the MS has transferred to the new cell the old DCK may be retained, or alternatively the authentication procedure shall be followed, and a new DCK generated.

4.2.3.2 Announced reselection type 3, unannounced and undeclared reselection

Where the new cell is not known, no key information can be transferred prior to cell reselection. The MS shall be required to register in the new cell, and the SwMI shall either then transfer the MS's existing DCK to the new cell, or require the MS to authenticate and establish a new DCK. Once the DCK is established, the CCK for the new cell shall be sent to the MS.

4.2.4 Continuity of ciphering at cell change

4.2.4.1 Cell change with uninterrupted ciphering

A cell change may be carried out with uninterrupted ciphering provided that a registration or authentication is not required to take place without encryption applied.

This can occur:

- on a change of connection within the same BS;
- on a change to another cell within the same LA;
- wherever key transfer is possible within the SwMI between cells in a protected manner.

If the MS is moving to a new LA, and protected key transfer can take place, announced reselection types 1 and 2 shall enable the MS to be given the CCK for the new cell before cell transfer. The new cell can also be sent the DCK of the MS.

If the MS is moving to a new LA , and protected key transfer can take place, announced reselection type 3, unannounced and undeclared reselection should enable the MS to register in encrypted mode. The new cell shall obtain the DCK of the MS by reference to its ITSI. The MS shall therefore attempt to register in encrypted mode; and shall only resort to unencrypted mode if the encrypted registration attempt fails.

4.2.4.2 Cell change with interrupted ciphering

A cell change shall require ciphering to be interrupted if the SwMI is not able to transfer keys between cells in a protected manner. The MS shall therefore either register or authenticate or both without encryption to generate a new DCK.

Registration should be avoided if announced reselection types 1 and 2 can occur; therefore only authentication should be needed to establish a DCK. The other types of reselection shall require a registration to occur also.

4.3 Signalling information confidentiality protocol

4.3.1 General

The signalling information confidentiality protocol shall be used to:

- start or stop the encryption service;
- identify the cipher KSG;
- identify the CK used;
- initiate the loading of the CK to the KSG;
- exchange the encryption mode control messages to synchronise encryption.

The protocol shall involve layers and sublayers of layer 3 (Mobility Management MM and Mobile Link Entity MLE), and of layer 2 (Logical Link Control LLC and MAC) of the TETRA protocol stack.

The security procedure in the MS shall be controlled by MM, which may indicate its security state to the MS application by the TNMM SAP. The application in the MS shall not however change the security state; this shall only be performed by the SwMI.

4.3.1.1 Positioning of encryption process

The encryption process itself shall be located in the upper part of the MAC layer, which itself is the lower part of layer 2. Situating the encryption process at this point, prior to channel coding at the transmitting end and after channel decoding at the receiving end, enables the MAC headers to be left unencrypted. This allows the appropriate channel coding to be used, and enables receiving parties to determine the applicability of a message received over air for them, and so enables them to apply the correct key for the decryption process. Figure 5 illustrates this interconnection:

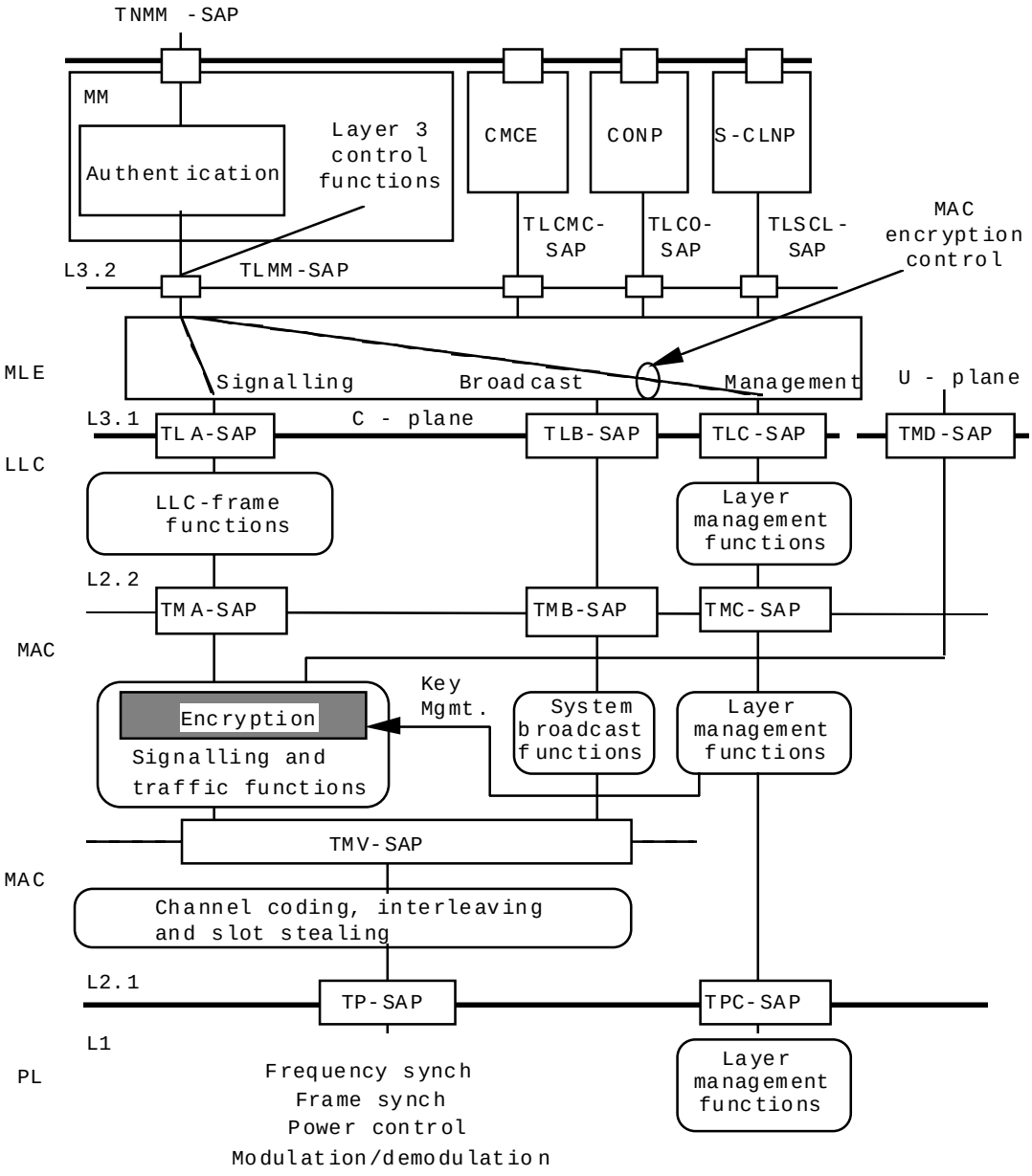


Figure 5: Relationship of security functions to layer functions

4.3.1.2 Operation of encryption process

The encryption process shall be controlled by MM.

At registration, ciphering parameters may be exchanged with the SwMI using the Location Update PDU exchange. Once keys have been established, MM may initiate the encryption process by passing the encryption parameters to the MAC by using the MLE_INFO_request primitive.

If rejection occurs due to a mismatch in ciphering parameters, MM may change the parameters and may try a new registration attempt. If the MS has only one set of ciphering parameters, MM shall attempt to establish a clear connection.

The SwMI may control the encryption process. In this case, PDUs shall be sent from the SwMI to the MS-MM forcing a re-registration procedure with new ciphering parameters set. If the re-registration is successful, MM shall pass the new encryption parameters to the MAC using the MLE_INFO_request primitive as before.

The encryption state shall not be changed without following the registration procedure. For optimum security, the authentication procedure should also be followed where implemented on a change of encryption state.

4.3.2 Service description and primitives

Each layer in the protocol stack provides a set of services to the layer above. This subclause describes the services that are added to those provided by each layer due to the incorporation of encryption, in addition to those specified in ETS 300 392-2 [2]. The primitives that are passed between the layers are also described.

4.3.2.1 Mobility Management (MM)

TNMM SAP: The encryption control procedure shall only be invoked by the SwMI using the registration procedure. The MS-MM may indicate its current state, or a change of state, to the MS application.

The following parameters shall be added to the TNMM-REGISTRATION -confirm and -indication primitives to enable indication of the encryption process in the MS:

- encryption control, on/off (mandatory) 1 bit;
- KSG number (Conditional) 4 bits;
- CK type, SCK/DCK (Conditional) 1 bit;
- Key Number (KN) (Conditional) 5 bits.

NOTE: The presence of the conditional bits depends on the state of the mandatory bit; i.e. the conditional fields are only present when encryption control is set to “on”.

4.3.2.2 Mobile Link Entity (MLE)

LMM SAP of MLE service boundary.

At the LMM SAP the following MLE services shall be provided to MM:

- loading of keys;
- start and stop ciphering.

These services shall be achieved by passing information to the MAC layer using MLE_INFO_request primitives.

The MLE_INFO_request primitive shall contain the following parameters to download keys:

- KSG number: (Mandatory) 4 bits;
- Key type: (Mandatory) 2 bits (SCK, DCK or CCK);
- Key number: (Mandatory) 5 bits (32 values of SCK or 2 values of CCK);
- Encryption key: (Mandatory) 80 bits;
- Alias Stream: (Optional) 24 bits (Only present when CCK is downloaded).

The MLE_INFO_request primitive shall contain the following parameters to control encryption:

- Encryption control, on/off (Mandatory) 1 bit;
- KSG number: (Conditional) 4 bits (Omitted if encryption turned off);
- Key type: (Conditional) 2 bits (Omitted if encryption turned off);
- Key number: (Conditional) 5 bits (Omitted if encryption turned off).

The CK type number shall determine the transmit mode only, and shall only take the value for SCK or DCK in an MS. In a BS, the selection shall be dependent on the associated MAC identity; it may be SCK or DCK to an individual subscriber, or SCK or CCK to a group. A CK shall be downloaded for each separate MAC identity.

4.3.2.3 LLC and MAC

The layer 2 service shall be to load keys and start and stop the ciphering as required by the MM/MLE request. The MAC shall also be responsible for applying the correct key depending on the identity placed in the header of each MAC PDU. This is described in ETS 300 392-2 [2]. The MAC shall also generate or decode the aliassed identity using the AS. The key selected for the encryption process shall be based on the identity without aliassing applied.

The encryption and aliassing processes shall be performed in the upper MAC before CRC, FEC and interleaving.

4.3.3 Protocol functions

Each functional entity in the protocol stack shall communicate with its peer entity using a defined protocol; for example the MM entity in the MS communicates with its peer MM entity in the SwMI. The incorporation of encryption at the air interface requires additional functions to be added to some of the functional entities of the protocol stack. These functions are described in this subclause.

4.3.3.1 MM

The protocol functions for air interface security shall be the following:

- ciphering type elements shall be contained in the U- and D- LOCATION UPDATE PDUs. A negotiation for ciphering types shall be performed in a re-registration if the parameters are not acceptable;

- MM shall perform a re-registration if the SwMI requires a change in the ciphering parameters including on-off control of encryption.

4.3.3.2 MLE

No encryption functionality shall be added to the MLE protocol.

4.3.3.3 LLC

No encryption functionality shall be added to the LLC protocol.

4.3.3.4 MAC

No encryption functionality shall be added to the MAC protocol.

4.3.4 PDUs for cipher negotiation

Ciphering elements shall be contained in the U_LOCATION_UPDATE-DEMAND, the D_LOCATION_UPDATE-COMMAND and the D_LOCATION_UPDATE-REJECT PDUs to permit negotiation of encryption parameters. These PDU’s are described in ETS 300 392-2 [2].

4.3.5 Protocol sequences

Figure 6 shows the protocol sequence in the normal case:

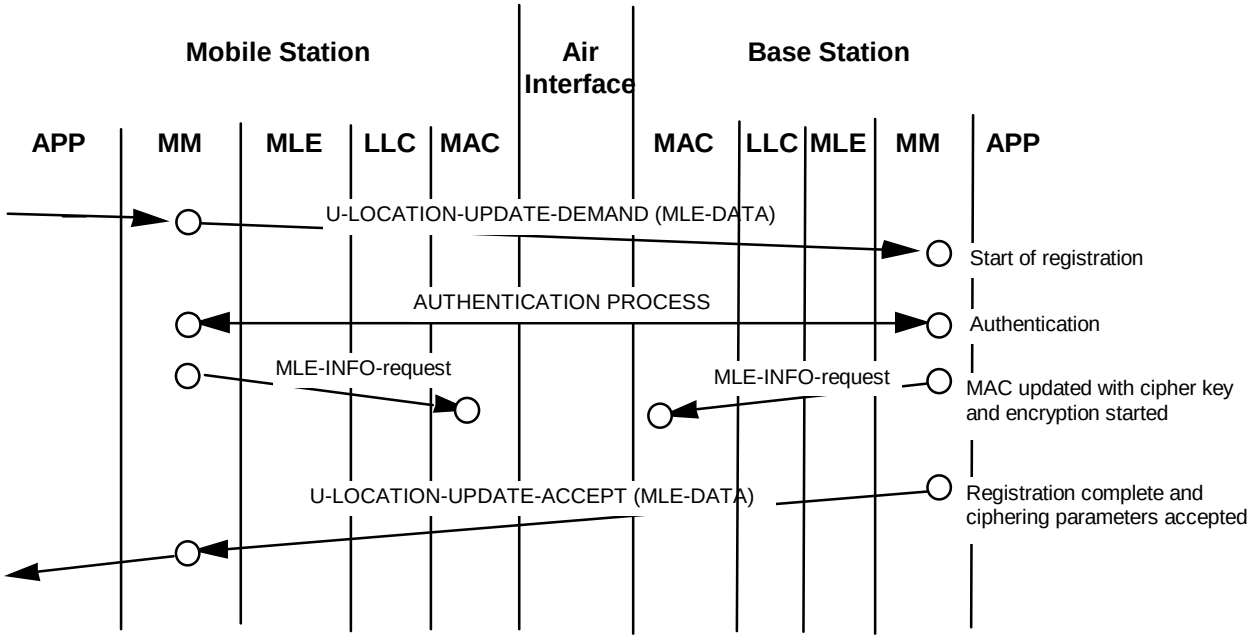


Figure 6: Protocol sequence

The protocol shall be the following:

1) When an MS camps on a cell, it shall attempt to register by sending a U_LOCATION_UPDATE_DEMAND PDU from MM. This shall contain the MS's chosen ciphering parameters.

2) If the registration parameters, including the security parameters, are acceptable, the authentication process as described in clause 5 shall be followed.

3) If the MS parameters are not acceptable to the SwMI, a D_LOCATION_UPDATE_REJECT shall be sent back to the MS. If it is a security parameter that is unacceptable, the MS may send in a new U_LOCATION_UPDATE_DEMAND with different parameters. Typically, if there is a key type, key number or KSG mismatch, the MS may retry with no encryption requested.

4) Once the authentication is complete, the MS and SwMI shall both send the ciphering information to their respective MAC layer processes using the MLE_INFO primitive. Once the MLE_INFO parameters are received in the MAC layer, encryption can be applied to all future messages requiring it.

5) The SwMI shall finally send a D_LOCATION_UPDATE_ACCEPT to the MS; this shall provide confirmation that the ciphering parameters suggested by the MS are acceptable.

There is a possibility that due to signalling failures in poor conditions, one party to the transaction may initiate ciphering before the other party. Any errors arising can be mitigated by the use of the MAC header to indicate whether the enclosed PDU is clear or encrypted: if one party is still attempting to re-send an earlier clear message, the state of the message shall be indicated to the other party and the decryption process shall not be applied as appropriate.

If the application in the SwMI requires a change of state of ciphering parameters, a D_LOCATION_UPDATE_COMMAND PDU shall be sent to the MS, with the ciphering parameters above determining the new requested ciphering state.

The MS shall respond with a U_LOCATION_UPDATE_DEMAND PDU with the appropriate ciphering parameters set. If the SwMI has commanded a change to a set of parameters not supported in the MS,

the MS may suggest that ciphering is stopped; or alternatively the MS shall not be able to re-register on that cell of that SwMI.

4.3.6 Key numbering and storage

Separate SCKs may be stored for each ITSI, a maximum of 32 keys may be stored.

5 Air Interface authentication and key management mechanisms

5.1 Security mechanisms

5.1.1 Requirements

The following requirements shall be taken into account:

- the mechanisms used to realise the authentication services and the key management functions shall be based on symmetric cryptographic algorithms. They assume that both parties share a common secret key, that has to be distributed before the authentication process. Authentication shall be successfully performed by proving the knowledge of the secret key to the other party;

- in order not to be vulnerable to a replay attack, the exact way of delivering this proof shall change unpredictably from instance to instance of an authentication;

- a synchronised real time clock shall not be required within the terminals by the authentication mechanism;

- the exchange of CKs shall be linked to the process. This shall ensure that the CK has indeed been exchanged with the party that has just been authenticated, and moreover shall extend the validity of the authentication from a point in time to the whole usage period of the CK;

- there shall be separate mechanisms for authentication of a user by the infrastructure and for authentication of the infrastructure by a user. It shall be possible to combine these two mechanisms to achieve mutual authentication of user and infrastructure;

- there shall be individual keys for each user to protect individually addressed traffic and a common key to protect messages in group calls;

- there shall be a key hierarchy of master and session authentication keys. There shall not be a necessity to store master keys in BSs or forwarded to visited networks;

- a spoofer who is able to compromise one of the BSs (could be an untrustworthy operator of a visited network) but not the central authentication module shall not be able to falsely authenticate himself as a legitimate user to any other but this particular BS and only then until a new session authentication key is chosen.

5.1.2 Authentication of a user

In this subclause, a mechanism is described that should be used to achieve the authentication of a user of a TETRA terminal by the TETRA infrastructure. This shall be done using a challenge response protocol, with a session authentication key derived from an authentication key that shall be shared by the user and the infrastructure. The session authentication key shall be provided by an authentication centre of the home system.

The computation of the session authentication key should be done by an algorithm called TA11. The computation of the response should be done by another algorithm TA12, which at the same time produces a derived CK.

The infrastructure shall produce a random number as a challenge called RAND1. The terminal shall compute a response called RES1 and the SwMI shall compute an expected response called XRES1. A derived CK shall be generated by this process, labelled DCK1. The entire protocol is summarised in figure 7.

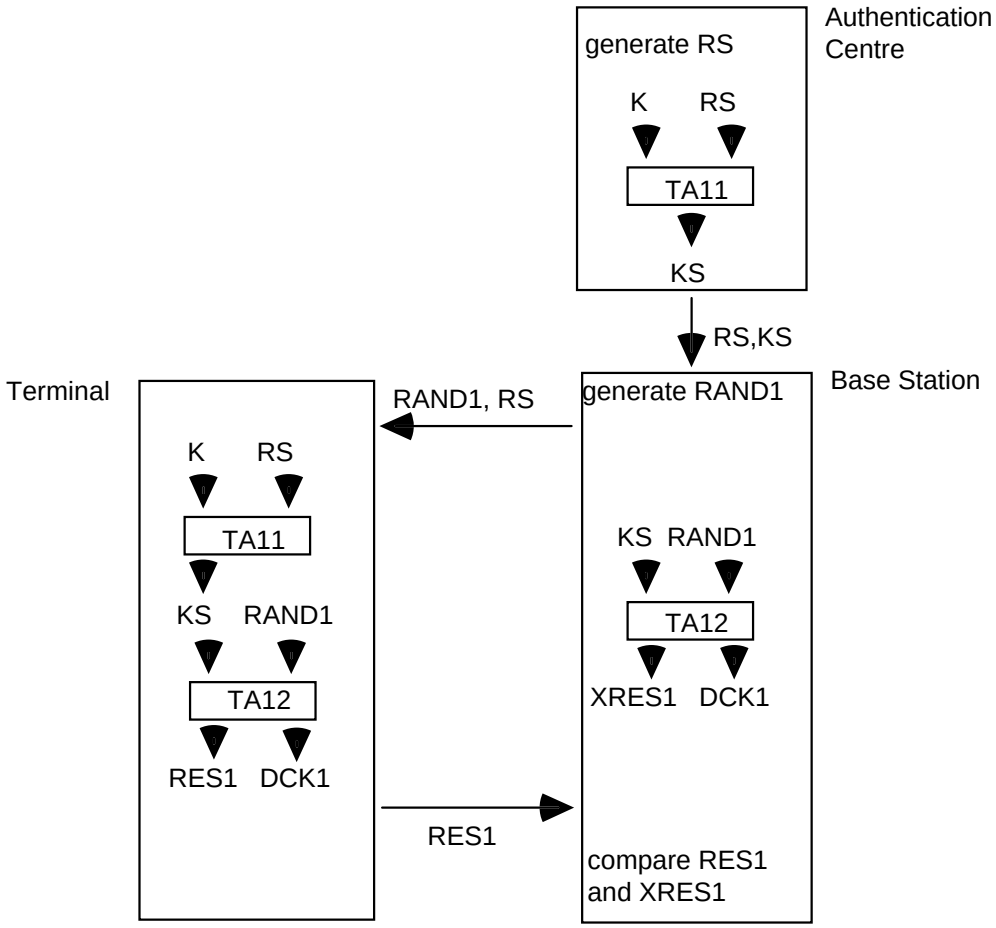


Figure 7: Authentication of a user by the infrastructure

5.1.3 Authentication of the infrastructure

Authentication of the infrastructure by a user shall be done in the same way as described in subclause 5.1.2 with the roles of the claimant and verifier reversed. The terminal shall generate a challenge called RAND2, the SwMI shall generate an actual response, and the terminal shall generate an expected response, called RES2 and XRES2 respectively.

The same authentication key K and random seed RS shall be used as in the case of authentication of the user by the infrastructure. However, the algorithms should be different: TA11 shall be replaced by TA21 and TA12 by TA22. Hence, there should also be a different value for the session authentication key, called KS'. The process is summarised in figure 8.

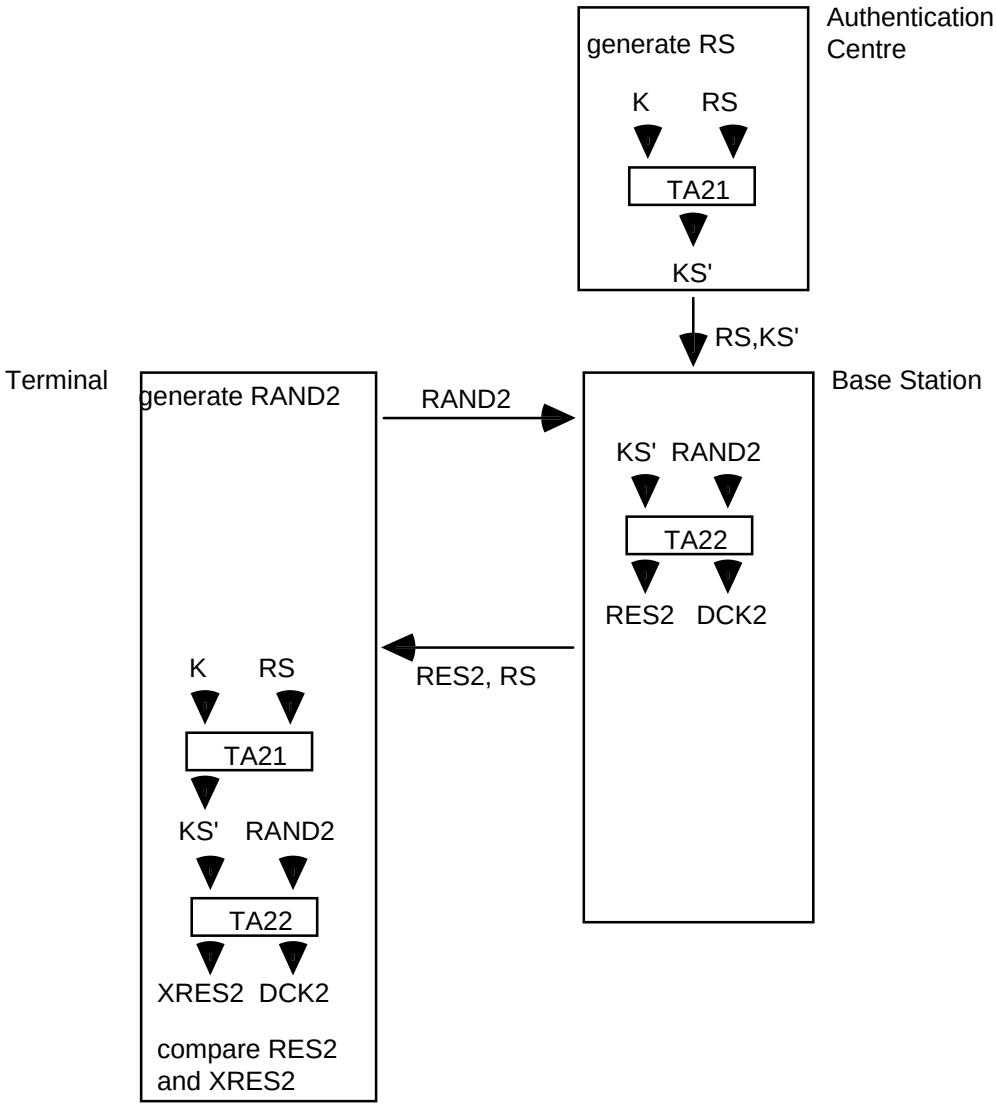


Figure 8: Authentication of the infrastructure by a user

5.1.4 Mutual authentication of user and infrastructure

Direct mutual authentication of user and infrastructure may be achieved by combining the two mechanisms described in subclauses 5.1.2 and 5.1.3.

Indirect mutual authentication shall be achieved if the derived CK is used for the encryption of the signalling data after a successful (unilateral) authentication of the user by the infrastructure or vice versa. Since neither side can derive the correct CK without the knowledge of the session authentication key KS, the correct working of the enciphered signalling shall be used to prove the knowledge of KS of either party.

5.1.5 Generation of the authentication key

Users should be authenticated by a process that is carried out in the terminal, as described in subclause 5.1.2. Therefore, the user should be able to control the authentication key. One way to exercise this control may be to enter the authentication key either directly or through a detachable module. The key may be stored in a module, detachable or not. Protection should be provided against the use of lost or stolen modules/terminals. This may be done by requiring the user to enter a Personal Identification Number (PIN).

The PIN may be checked in either of two ways. It may either be checked locally against a reference stored in the module or it may be used to generate the authentication key in conjunction with other data stored within the module. In the latter case the actual checking shall be carried out by the infrastructure.

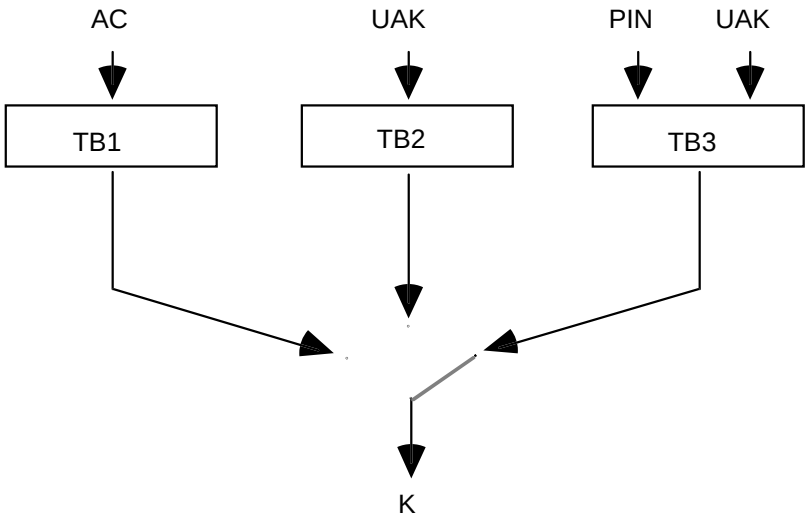


Figure 9: Generation of the authentication key

There shall be three different cases for the generation of the authentication key, which are summarised in figure 9:

1) the authentication key may be generated from an authentication code AC that is manually entered by the user. In this case the code shall be remembered by the user and should not normally be longer than a few digits. The procedure to generate the authentication key from AC is labelled TB1;

2) the authentication key may be generated from a user authentication key UAK stored in a module (detachable or not). In this case the UAK can be a random value of a desirable length (e.g. 128 bits). However the user should be required to enter a PIN that can be checked before the UAK can be used for authentication. The checking of the PIN should not be part of the mechanism and, therefore, not shown in figure 9. The procedure to generate the authentication key from UAK is labelled TB2;

3) the authentication key may be generated from both the UAK stored in a module and the PIN entered by the user. The procedure to generate the authentication key from UAK and PIN is labelled TB3.

5.1.6 CKs

5.1.6.1 The DCK

As explained in subclauses 5.1.2 and 5.1.3, successful authentication of the user or the infrastructure shall result in the generation of a derived CK DCK1 or DCK2, respectively. The DCK should therefore be derived from its two parts DCK1 and DCK2 by the procedure TB4, as shown in figure 10.

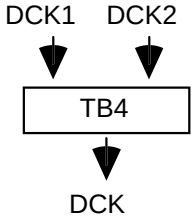


Figure 10: Derivation of the DCK from its two parts

In case of unilateral authentication, either DCK1 or DCK2 shall be set to zero.

DCK may be used to protect voice, data, and signalling sequences between the infrastructure and an individual terminal after successful authentication has taken place.

5.1.6.2 The SCK

A static CK SCK shall be a fixed value that should be known to the infrastructure and every terminal. It may be used to give some limited amount of security to signalling data that is sent before authentication has taken place or after it has expired.

5.1.6.3 The CCK

The authentication key shall be individual to each user. Therefore, the DCK that is derived from it should also be individual. It, therefore, cannot be used in a group call, because a message is only broadcast once and has to be deciphered by each group member within a particular location area. A common CK CCK should be used to provide confidentiality for these messages.

The CCK shall be generated by the infrastructure and distributed to the terminals. There shall be one such key for every LA. A terminal should be provided with the CCK after successful authentication. The CCK may then be transmitted in encrypted form using algorithm TA31 and the derived CK DCK. To allow the CCK to be decrypted by the terminal, algorithm TA31 shall have an inverse TA32. To allow the terminal to discover if CCK has been corrupted due to transmission errors or manipulation, TA31 may introduce some redundancy into the sealed common CK SCCK. The redundancy should be checked by TA32. A detected manipulation shall be indicated by setting the manipulation flag MF.

The infrastructure may change the CCK and distribute the new key to the terminals. For this purpose, a key number KN shall be encrypted and distributed along with the key. The key number shall be incremented for each new key and shall be referenced by one bit in the header of the encrypted message to select the active CCK. The value of this bit shall equal the value of the least significant bit of KN. By checking that the key number of a newly distributed CCK has been increased, the terminal may protect itself against replay of old keys.

5.1.6.4 The AS

An alias stream AS shall also be associated with a CCK, and used to generate an alias for each subscriber identity transmitted across the air interface. This shall also be encrypted and distributed with the CCK. A description of the AS is given in subclause 4.1.6. The process is summarised in figure 11.

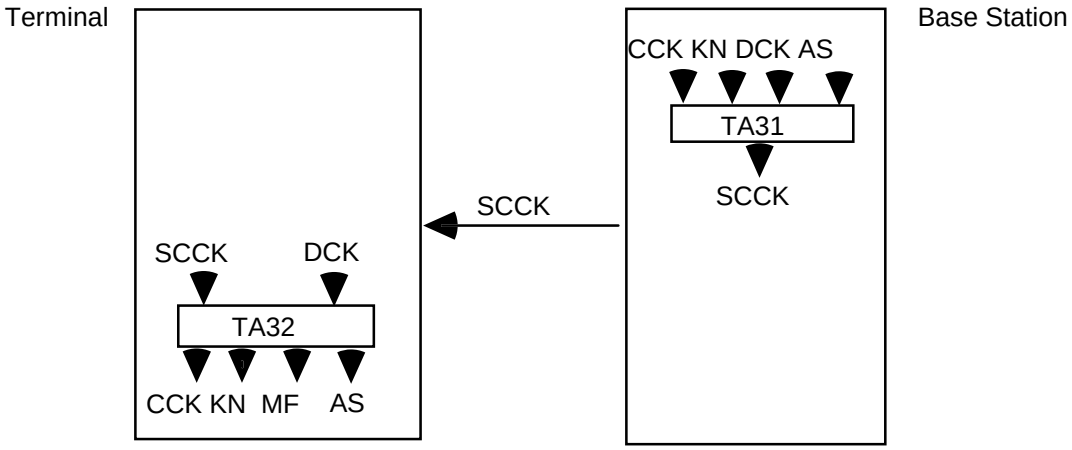


Figure 11: Distribution of a common CK

5.2 Definition of protocols

The air interface authentication protocol shall involve layer 3 MM.

5.2.1 Service description and primitives

5.2.1.1 Authentication service

At the TNMM SAP, a specific service shall be provided to allow an application to initiate an authentication exchange and to receive its result. The MS application must also respond to an authentication demand from the SwMI. The primitives required shall be as follows:

- TNMM-MS_AUTHENTICATE indication shall be used to indicate to the MS application a demand for authentication received from the SwMI;

- TNMM-MS_AUTHENTICATE response shall be used to provide the appropriate response to the challenge received in the demand for authentication;

- TNMM-MS_AUTHENTICATE confirm shall be used to confirm successful or failed authentication of the MS by the SwMI;

- TNMM-BS_AUTHENTICATE request shall be used by the MS application to initiate an authentication of the SwMI;

- TNMM-BS_AUTHENTICATE indication shall be used to report to the MS application the response returned by the SwMI, and the random seed to be used for DCK generation;

- TNMM-BS_AUTHENTICATE response shall be used by the MS application to signal success or failure of the authentication to the SwMI.

Table 2: TNMM AUTHENTICATE service primitives

Generic name	Specific name	Parameters
TNMM-MS_AUTHENTICATE	indication	random challenge, random seed
TNMM-MS_AUTHENTICATE	response	value, mutual authentication flag
TNMM-MS_AUTHENTICATE	confirm	result
TNMM-BS_AUTHENTICATE	request	random challenge
TNMM-BS_AUTHENTICATE	indication	response value, random seed, mutual authentication flag
TNMM-BS_AUTHENTICATE	response	result

5.2.1.2 CCK distribution and generation service

Another service shall be provided to allow an application to initiate the generation and distribution of a new common CK. The primitives required shall be as follows:

- TNMM-NEWKEY request shall be used to request the distribution of a new common CK. It shall contain details of the location area for which the key is requested, and the last key number (if any) held for that location area;

- TNMM-NEWKEY confirm shall be used to provide the MS application with the new common CK and associated parameters;

- TNMM-NEWKEY response shall be used by the MS application to confirm that the key information received is acceptable.

Table 3: TNMM NEWKEY service primitives

Generic name	Specific name	Parameters
TNMM-NEWKEY	request	last key number, location area ID
TNMM-NEWKEY	confirm	Sealed Common CK (SCCK), key number, location area ID
TNMM-NEWKEY	response	Result

5.2.2 Protocol functions

An authentication exchange can be requested, either explicitly or as part of the registration procedure. It can be initiated by the MS or SwMI side. The initiating side shall send an Authentication Demand PDU that shall always be answered by the other side with an Authentication Response PDU. Success or failure of the authentication shall be communicated either by a specific Authentication Result PDU or as part of the registration procedure.

After a successful authentication exchange, both MS and SwMI shall update the relevant part of the derived CK, DCK1 or DCK2, and the derived CK DCK accordingly.

After a successful authentication or after generation of a new common CK, one or several of the most recently generated CCKs may automatically be distributed to the MS.

5.2.3 Protocol Data Units (PDUs)

D-Authentication Demand: Shall be used by the infrastructure to start the authentication procedure of a user.

Direction: SwMI to MS;
Service used: MM;
Response to: U-Location Update Demand or none;
Response expected: U-Authentication Response.

Table 4: D-Authentication Demand PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Random challenge	80	M		
Random seed	80	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

U-Authentication Response: Shall be used by the MS to reply to a demand message. The “Mutual Authentication Flag” shall be used to indicate to the infrastructure whether it should expect a U-Authentication Demand message after sending the D-Authentication Result.

Direction: MS to SwMI;
Service used: MM;
Response to: D-Authentication Demand;
Response expected: D-Authentication Result.

Table 5: U-Authentication Response PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Response value	32	M		
Mutual authentication flag	1	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

D-Authentication Result: Shall be used by the infrastructure to notify the MS about the result of the authentication exchange.

Direction: SwMI to MS;
Service used: MM;
Response to: U-Authentication Response;
Response expected: U-Authentication Demand or none (depending on whether the Mutual Authentication Flag was set).

Table 6: D-Authentication Result PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Authentication result	1	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

U-Authentication Demand: Shall be used by the MS to start the authentication procedure of the infrastructure.

Direction: MS to SwMI;
Service used: MM;
Response to: D-Authentication Result or none;
Response expected: D-Authentication Response.

Table 7: U-Authentication Demand PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Random challenge	80	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

D-Authentication Response: Shall be used by the SwMI to reply to a demand message. The “Mutual Authentication Flag” shall be used to indicate to the terminal whether it should expect a D-Authentication Demand message after sending the U-Authentication Result.

Direction: SwMI to MS;
Service used: MM;
Response to: U-Authentication Demand;
Response expected: U-Authentication Result.

Table 8: D-Authentication Response PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Response value	32	M		
Random seed	80	M		
Mutual Authentication Flag	1	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

U-Authentication Result: Shall be used by the MS to notify the SwMI about the result of the authentication exchange.

Direction: MS to SwMI;
Service used: MM;
Response to: D-Authentication Response;
Response expected: D-Commonkey Provide or D-Authentication Demand (depending on whether the Mutual Authentication Flag was set).

Table 9: U-Authentication Result PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Authentication result	1	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

U-Commonkey Demand: Shall be used by the MS to demand the distribution of the currently common CKs from the infrastructure.

Direction: MS to SwMI;
Service used: MM;
Response to: none;
Response expected: U-Commonkey Provide.

Table 10: U-Commonkey Demand PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Last key number	16	M		
Location area ID	14	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

D-Commonkey Provide: Shall be used by the SwMI to distribute common CKs to the MS.

Direction: SwMI to MS;
Service used: MM;
Response to: U-Authentication Result or U-Commonkey Demand or none;
Response expected: U-Commonkey Accept.

Table 11: D-Commonkey Provide PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Sealed common CK	144	M		
Key number	16	M		
Location area ID	14	M		
More keys indicator	1	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

U-Commonkey Accept: Shall be used by the MS to indicate acceptance of new common CKs.

Direction: MS to SwMI;
Service used: MM;
Response to: D-Commonkey Provide;
Response expected: D-Commonkey Provide or D-Location Update Accept or none.

Table 12: U-Commonkey Accept PDU contents

Information element	Element length	Element type	Reference	Remark
Message identifier	4	M		
Key number	16	M		
Location area ID	14	M		
More keys indicator	1	M		

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

5.2.4 Protocol sequences

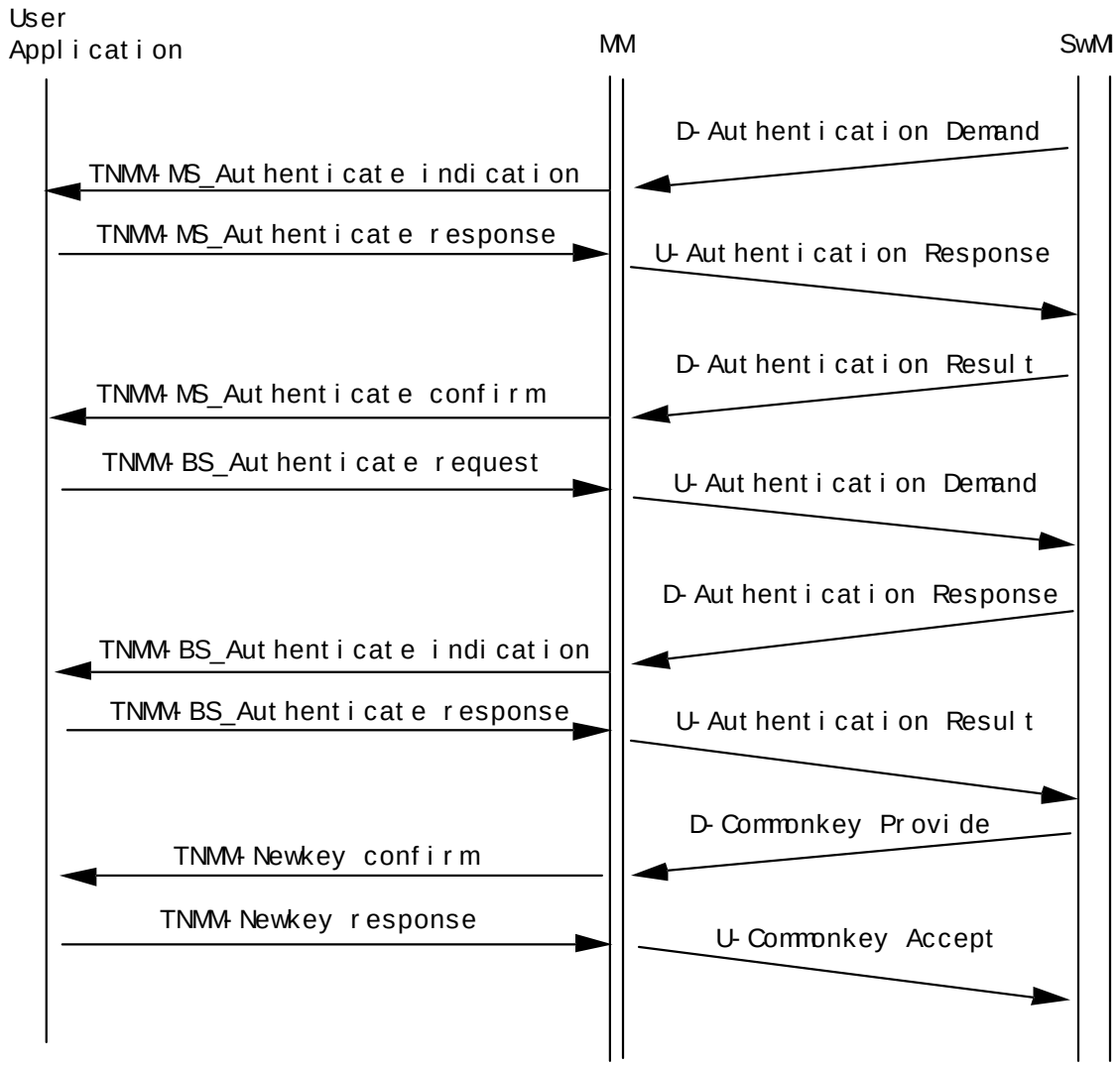


Figure 12: Protocol sequence of authentication without registration

Figure 12 shows the protocol sequence for mutual authentication in the case where the process is started by the infrastructure, not within the context of a registration procedure. Figure 13 shows the protocol sequence within the context of a registration procedure. In each case, the last two passes shall be repeated as often as necessary to distribute the necessary number of common CKs.

The figures show the most general and complete examples in the sense that mutual authentication and distribution of CCK is involved. By deleting the respective messages, sub-sequences can be generated that apply to more special cases.

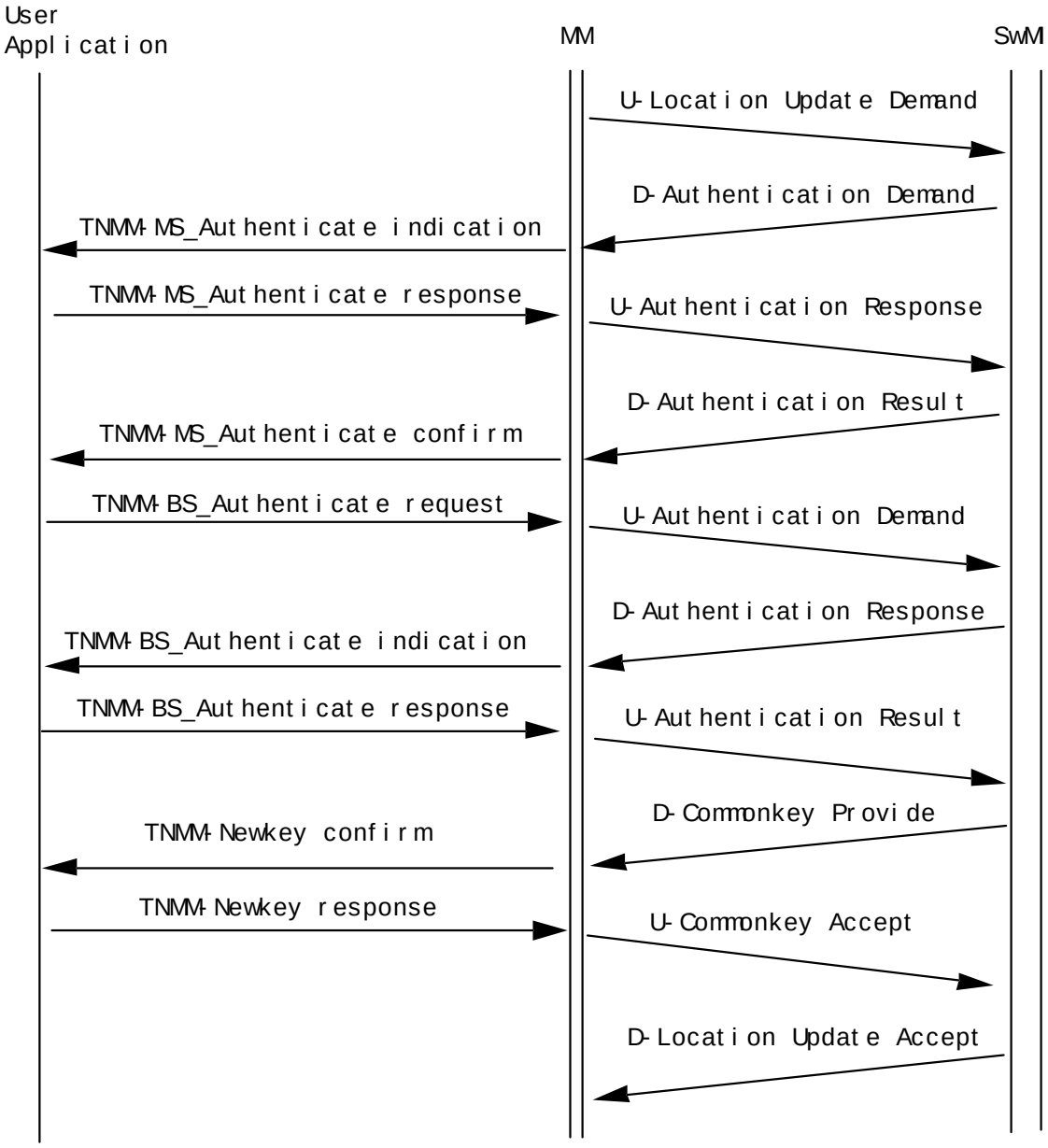


Figure 13: Protocol sequence of authentication registration

5.3 Boundary conditions for the cryptographic algorithms and procedures

In the following the symbol |XYZ| shall be used to denote the length of the parameter XYZ. If the length of a parameter can vary, |XYZ| denotes the range between the shortest and the longest possible values for XYZ.

TA11: Shall be used to compute the session authentication key KS from the authentication key K and the random seed RS. The algorithm shall have the following properties:

Input 1: Bit string of length |K|;
Input 2: Bit string of length |RS|;

Output: Bit string of length |KS|.

The algorithm should be designed such that it is difficult to infer any information about Input 1 from the knowledge of Input 2 and the Output (even if the details of the algorithm are known).

TA21: Shall be used to compute the session authentication key KS' from the authentication key K and the random seed RS. The algorithm shall have the following properties:

Input 1: Bit string of length |K|;
Input 2: Bit string of length |RS|;

Output: Bit string of length |KS'|.

The algorithm should be designed such that it is difficult to infer any information about Input 1 from the knowledge of Input 2 and the Output (even if the details of the algorithm are known).

TA12: Shall be used to compute the (expected) response (X)RES1 as well as the derived CK DCK1 from the session authentication key KS and the challenge RAND1. The algorithm shall have the following properties:

Input 1: Bit string of length |KS|;
Input 2: Bit string of length |RAND1|;

Output 1: Bit string of length |(X)RES1|;
Output 2: Bit string of length |DCK1|.

The algorithm should be designed such that it is difficult to infer any information about Input 1 or Output 2 from the knowledge of Input 2 and Output 1 (even if the details of the algorithm are known).

TA22: Shall be used to compute the (expected) response (X)RES2 as well as the derived CK DCK2 from the session authentication key KS' and the challenge RAND2. The algorithm shall have the following properties:

Input 1: Bit string of length |KS'|;
Input 2: Bit string of length |RAND2|;

Output 1: Bit string of length |(X)RES2|;
Output 2: Bit string of length |DCK2|.

The algorithm should be designed such that it is difficult to infer any information about Input 1 or Output 2 from the knowledge of Input 2 and Output 1 (even if the details of the algorithm are known).

TA31: Shall be used to compute the sealed common CK SCCK from the common CK CCK, alias stream AS, key number KN and the derived CK DCK. The algorithm shall have the following properties:

Input 1: Bit string of length |CCK|;
Input 2: Bit string of length |KN|;
Input 3: Bit string of length |DCK|;
Input 4: Bit string of length |AS|;

Output: Bit string of length |SCCK|.

The algorithm should be designed such that it is difficult to infer any information about Input 1 of Input 4 from the knowledge of Input 2 and the Output, provided that Input 3 is unknown (even if the details of the algorithms are known).

TA32: Shall be used to compute the common CK CCK, alias stream AS, and the key number KN from the sealed common CK SCCK and the derived CK DCK. The algorithm shall have the following properties:

Input 1: Bit string of length |SCCK|;
Input 2: Bit string of length |DCK|;

Output 1: Bit string of length |CCK|;
Output 2: Bit string of length |KN|;
Output 3: Boolean;
Output 4: Bit string of length |AS|.

The algorithm should be designed such that it is difficult to find for a fixed Input 1 a value for Input 2 that results in Output 3 assuming the value “FALSE”, provided that Input 2 is unknown (even if the details of the algorithms are known). Moreover, it shall be difficult to derive (parts of) Input 2 from the observation of various matching values of other inputs and outputs (known plain text attack).

TB1: Shall be used to compute an authentication key K from the authentication code AC. The algorithm shall have the following properties:

Input: Bit string of length |AC|;

Output: Bit string of length |K|.

The algorithm should be designed such that the Output is dependent on every bit of the Input.

TB2: Shall be used to compute an authentication key K from the user authentication key UAK. The algorithm shall have the following properties:

Input: Bit string of length |UAK|;

Output: Bit string of length |K|.

The algorithm should be designed such that the Output is dependent on every bit of the Input.

TB3: Shall be used to compute an authentication key K from the user authentication key and the personal identification number PIN. The algorithm shall have the following properties:

Input 1: Bit string of length |PIN|;
Input 2: Bit string of length |UAK|;

Output: Bit string of length |K|.

The algorithm should be designed such that the Output is dependent on every bit of both Inputs.

TB4: Shall be used to compute the derived CK DCK from the derived CK parts DCK1 and DCK2. The algorithm shall have the following properties:

Input 1: Bit string of length |DCK1|;
Input 2: Bit string of length |DCK2|;

Output: Bit string of length |DCK|.

The algorithm should be designed such that the Output is dependent on every bit of both Inputs.

5.4 Dimensioning of the cryptographic parameters

Table 13: Dimensioning of cryptographic parameters

Abbreviation	Name	No. of bits
AC	Authentication Code:	16 - 32 bits
AS	Alias Stream	24 bits
CK	Cipher Key:	80 bits
CCK	Common CK:	80 bits
DCK1	Part 1 of the derived CK:	80 bits
DCK2	Part 2 of the derived CK:	80 bits
DCK	Derived CK:	80 bits
MF	Manipulation flag:	Boolean
K	Authentication key:	128 bits
KN	Key Number:	16 bits
KS	Session authentication Key:	128 bits
KS'	Session authentication Key:	128 bits
PIN	Personal Identification Number:	16 - 32 bits
RAND1	Random challenge 1:	80 bits
RAND2	Random challenge 2:	80 bits
RES1	Response 1:	32 bits
RES2	Response 2:	32 bits
RS	Random Seed:	80 bits
SCK	Static CK:	80 bits
SCCK	Sealed Common CK:	144 bits
UAK	User authentication key:	128 bits
XRES1	Expected response 1:	32 bits
XRES2	Expected response 2:	32 bits

5.5 Summary of the cryptographic processes

Figure 14 gives a summary of the authentication mechanisms explained in the previous subclauses.

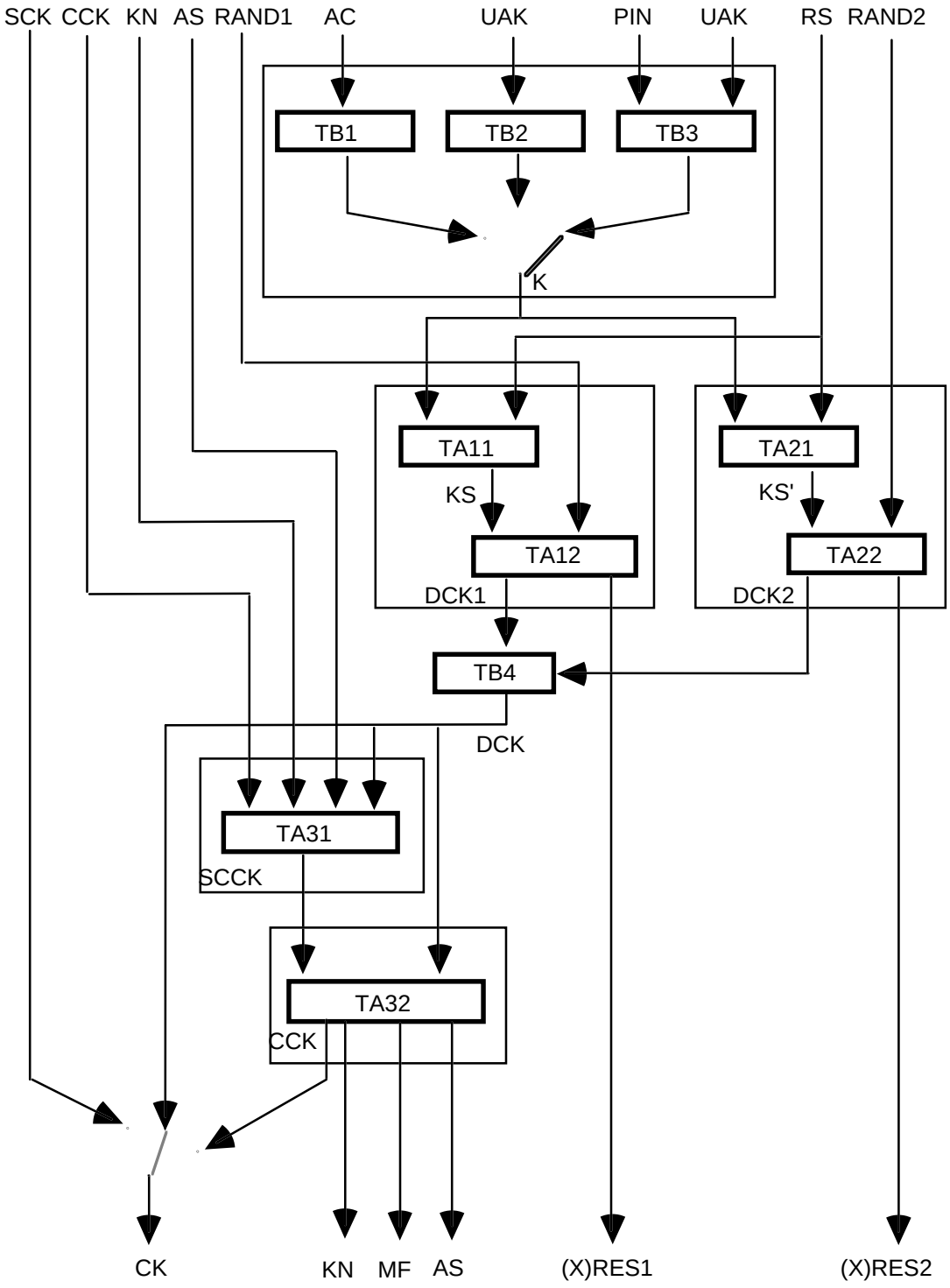


Figure 14: Overview of air interface authentication and key management

6 End-to-end encryption

6.1 Encryption mechanisms

This clause describes the encryption mechanisms, and the mechanism used to synchronise the encryption system. The method described here makes use of the Stealing Channel, STCH/E, as described in ETS 300 392-2 [2] for synchronisation.

6.1.1 Voice encryption mechanism

A functional diagram of the voice encryption mechanism based on the synchronous stream cipher principle is given in figure 15. Function F_1 shall combine the Plain Text (PT) and the KSS resulting in the Cipher Text (CT). On the receiving side the reverse process shall take place. This is illustrated in figure 16.

The KSG (TA) shall have two inputs, the derived key (DK) and the initialisation value (IV). IV is a random number that shall initialise the pseudo random number generator inside TA and may be generated inside the encryption unit or may come from outside the encryption unit. TA may be a standard or proprietary algorithm. TA should be selected by the algorithm selection value (AN). figure 17 gives the general structure of TA.

The derived key DK can be the result of a combination of a CK and the Call-ID or a replay offset value (RO). This shall be selected by the replay offset select signal (ROS). For more details refer to subclause 6.1.1.2.

The CK shall be selected by the key selection value (KN). New CKs can be loaded over the “control” interface.

In order to synchronise the KSG in the receiving terminal, a Synchronisation Value (SV) that is generated by the pseudo random number generator inside TA shall be transmitted to the receiving terminal(s). Because serially transmitting SV to the receiver shall require 1 half-slot time, the transmitter should advance its value of SV by one half-slot before transmitting SV, thus the value received at the receiver should be synchronised with the expected value at the start of the next half-slot.

SV shall be transmitted to the receiving terminal(s) in a synchronisation frame (half-slot) (SF) together with the values AN, KN and ROS for automatic algorithm, key and replay offset selection respectively by means of stealing frames (half-slots) from the encrypted PT. The frame stealing shall be performed by function F_2. In the receiving terminal function F_3 shall extract SF from CT. Refer to subclause 6.2 for a description of this frame stealing mechanism.

SF may also be used to transmit data to synchronise clocks for time stamping. The format of SF is given in subclause 6.2.5.

“Sync-control” shall be an autonomous device that determines when a new SF is transmitted. “Sync-detect” shall determine when an SF is received and whether it is received correctly or not. For details refer to subclause 6.2.

“Crypto-control” shall control and manage the encryption and keys. For example algorithm and key selection, derivation of DK and reporting of encryption status shall be functions of the “crypto-control”.

Apart from what has already been mentioned “crypto-control” shall have the following additional in- and outputs:

- Encryption Switch (ES): shall switch the encryption on and off. It may not be necessary to use ES as other parts of the U-plane may prevent unencrypted text from being routed through the encryption unit;

- Key Command (KC): shall indicate what should be done with the key data;

- Transmit Status (TRS): shall indicate that the transmission has started and that initial synchronisation should begin;

- Reception Status (RCS): shall indicate when reception has started or stopped;

- User Device (UD): shall be used as addressing information to associate call and encryption data with an individual entity beyond the encryption unit;

- Traffic Channel type (TCH): shall indicate the type of traffic channel i.e. speech or data at the various rates;

- Slot Rate (SR): shall give the number of slots per frame;

- Cipher Report (CR): shall be used to acknowledge keys and provides information about the synchronisation status;

- Cipher Identifier (CID): shall be used to identify the encryption unit;

- Stolen Half-Slot Identifier (SHSI): shall identify the meaning of the received stolen half-slot;

- Synchronisation Status (SS): shall indicate to “crypto-control” whether the receiver is in synchronisation or out of synchronisation.

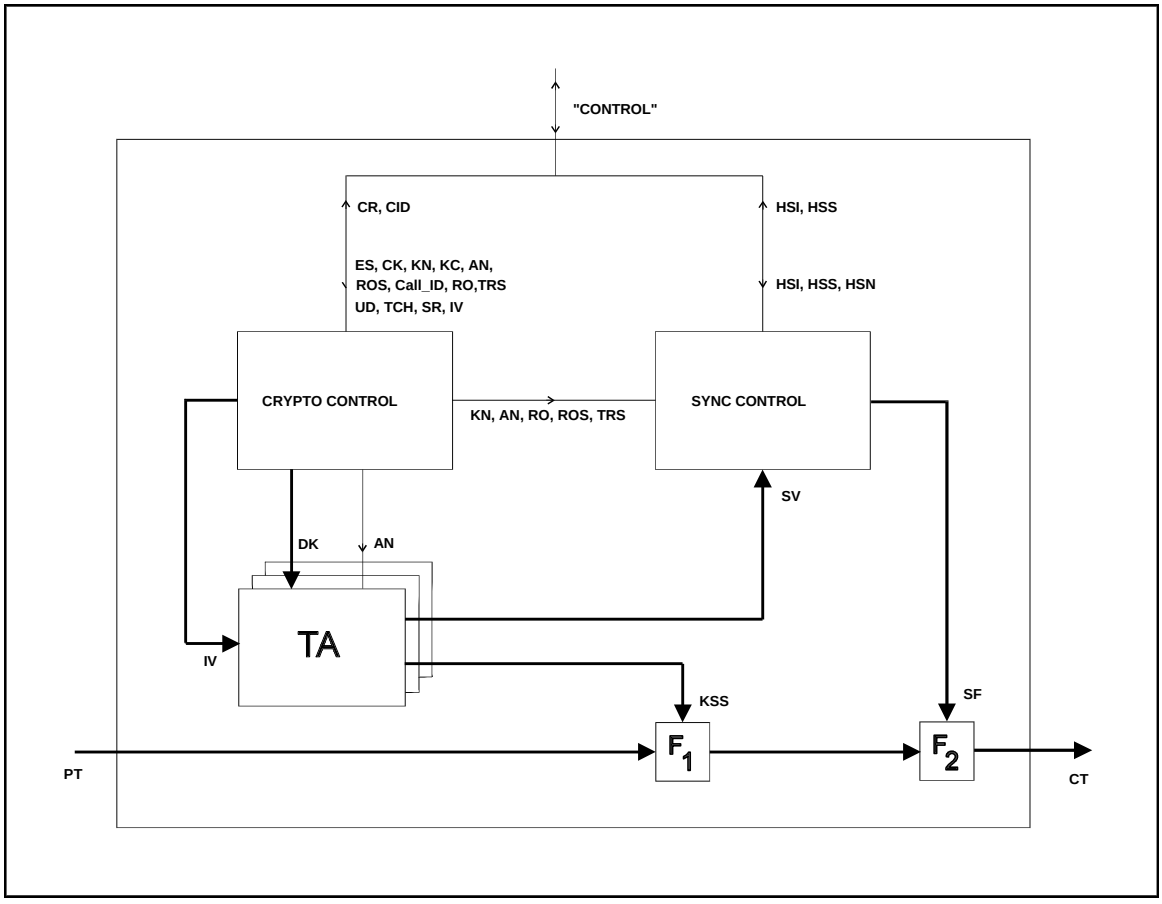


Figure 15: Functional diagram voice encryption mechanism

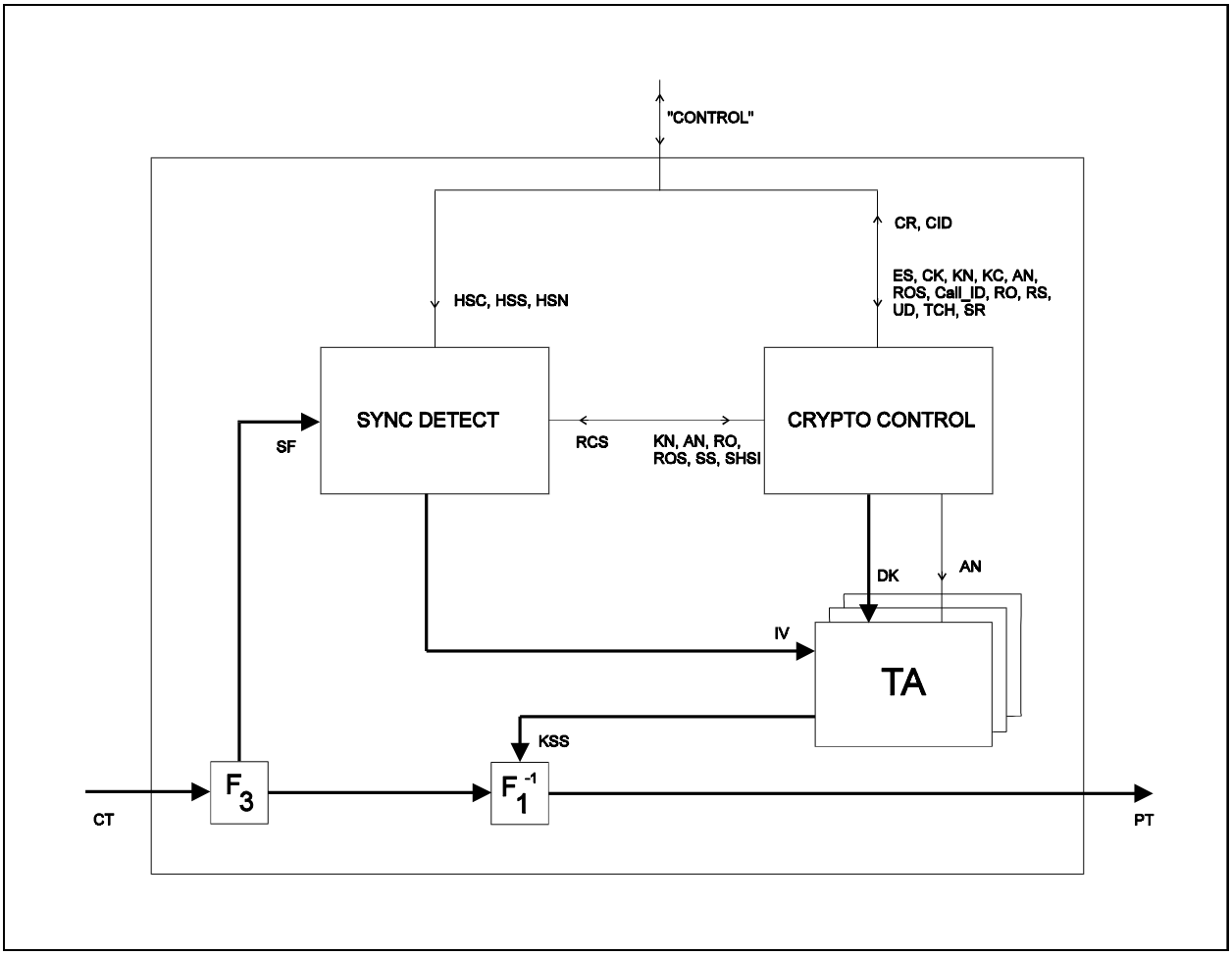


Figure 16: Functional diagram voice decryption mechanism

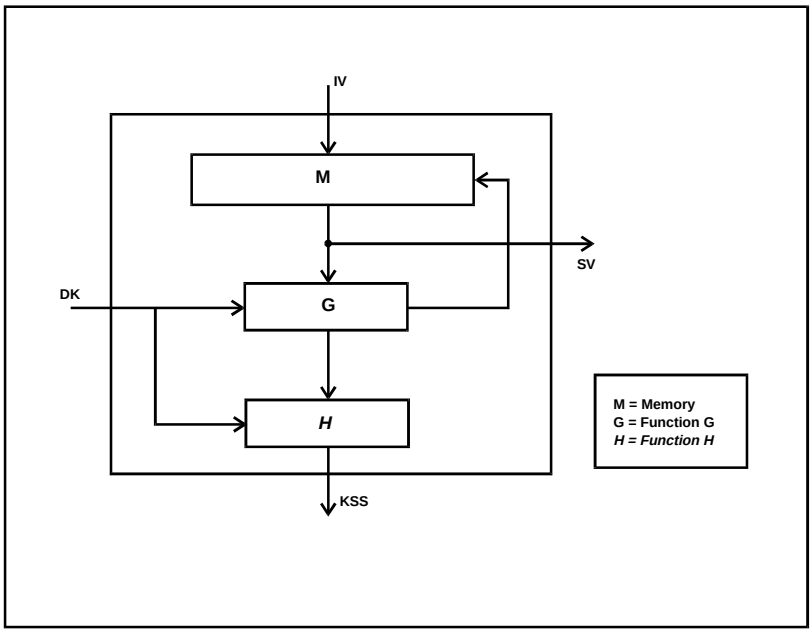


Figure 17: General structure of TA

6.1.1.1 Description of functions

6.1.1.1.1 Functions F_1 and F_1^{-1}

Function F_1 shall combine the bit streams PT and KSS resulting in an encrypted bit stream CT. Function F_1^{-1} shall be the inverse of F_1 and shall combine the bit streams CT and KSS resulting in the decrypted bit stream PT.

An example of a possible implementation follows. If PTS[k], KSS[k] and CTS[k] are the k-th. bits of the bit stream segments PTS, KSS and CTS respectively, where k=0 corresponds with the first output bit in the bit stream segments, then

function F_1: CTS[k] = PTS[k] + KSS[k] (mod 2) for k = 0,..,LEN-1

function F_1^{-1}: PTS[k] = CTS[k] + KSS[k] (mod 2) for k = 0,..,LEN-1

where LEN is the length of the bit stream segments.

6.1.1.1.2 Function F_2

Function F_2 shall replace a half slot of encrypted plain text with a synchronisation frame provided by the “sync control”.

6.1.1.1.3 Function F_3

Function F_3 shall recognise the synchronisation frames in the received cipher text, and shall supply them to “sync detect”.

6.1.1.2 Protection against replay

Protection against replay shall be obtained by using a regularly changing DK for key stream generation. For this purpose DK may be the result of a combination of the selected CK and the Call-ID or RO. Which value is used for replay protection, shall be selected by ROS.

The Call-ID may be used to give a basic level of protection against replay. For a description of the Call-ID refer to ETS 300 392-2 [2]. The Call-ID may be changed during a call if the call duration exceeds a certain period or for other reasons.

An example of a possible implementation using the Call-ID follows. If Call-ID[k], CK[k] and DK[k] are the k-th. bits of the bit strings Call-ID, CK and DK respectively, where k=0 corresponds with least significant bit, then

DK[k] = CK[k] + Call-ID[k] (mod 2) for k = 0,..,13

DK[k] = CK[k] for k = 14,..,127

For more secure protection against replay RO may be used. RO may be a counter value or a time stamp based on a real time clock. The latter shall require an accurate local clock in every mobile terminal and a mechanism for synchronising these clocks on a regular base. Implementation of the counter, the clock, the synchronisation mechanism of the clock and the quantisation of the RO are outside the scope of this ETS.

6.1.2 Data encryption mechanism

Encryption of circuit mode data preferably should be implemented in the application requiring transport of data. However encryption of circuit mode data can also be achieved by using the voice encryption mechanism.

Using the voice encryption mechanism can only gain confidentiality. In order to achieve data integrity other precautions should be taken.

6.2 Synchronisation of encryption units

Synchronisation of the synchronous stream cipher in the encryption units in the transmitting and the receiving terminals shall require transmission of synchronisation data from transmitter to receiver. Synchronisation can be subdivided into an initial and a late-entry part. Both parts shall use frame stealing for transferring the synchronisation data.

Synchronisation shall be performed by “sync-control” in the transmitter and “sync-detect” in the receiver. “Sync-control” shall autonomously determine the points of time of transmitting synchronisation data. “Sync-detect” shall detect the received synchronisation and shall provide for a flywheel mechanism in case of wrongly received synchronisation data.

This clause successively describes the frame stealing mechanism, transmission of initial and late-entry synchronisation, reception of synchronisation and the stolen frame format.

6.2.1 Frame stealing

The transfer of synchronisation data in case of frame stealing shall be achieved by stealing speech frames (half-slots) from the codec output in the U-plane. The synchronisation frames shall be transmitted as individual half-slots via the stealing-channel STCH/E for initial as well as for the late-entry synchronisation.

When the encryption unit steals a frame it shall set the Half-Slot Importance (HSI) to HIGH.

A Half-Slot Stolen (HSS) indication shall be associated with each speech frame of a pair making up a transmission slot. The valid combinations shall be:

- neither Half-Slot Stolen;
- the first Half-Slot Stolen;
- both Half-Slots Stolen;
- second Half-Slot Stolen, only if this is the first half-slot available to the U-plane at the start of transmission.

HSS shall indicate to the codec and to the MAC that a speech frame has been stolen.

The Half-Slot Number (HSN) shall be incorporated in the data coming from the codec.

The encryption unit shall not steal U-plane signalling from the end user device. Half-slots stolen prior to the encryption unit shall be encrypted end-to-end (for example comfort noise messages).

Half-Slot Stolen by Encryption unit (HSSE) shall indicate to the receiver whether the frame is stolen by the encryption unit or not. A Stolen Half-Slot Identifier (SHSI) shall give the meaning of the data in the stolen frame. HSSE and SHSI shall be incorporated in SF. HSSE and SHSI shall not be encrypted, whether the remaining contents of SF are encrypted or not. The remainder of SF shall be encrypted unless the half slot contains synchronisation information.

The stealing-channel STCH/E shall allow 121 bits of data. For the format of SF refer to subclause 6.2.5.

In case of circuit mode data, when a half-slot is stolen, the data port (see Figure 19) should repeat the data in the half-slot which was stolen.

6.2.2 Transmission of initial synchronisation

Up to 4 high-importance half-slots may be stolen during the first 1-second of transmission to convey the initial synchronisation data. The distribution of the slots may be varied; they may be placed consecutively at the start of the transmission, before any speech is transmitted, or may be well spaced, with only a single Half-Slot Stolen before speech transmission commences.

The first SV transmitted at the start of each transmission shall be equal to IV, the random number that initialises TA in the transmitter. The SV's transmitted after that on behalf of initial synchronisation shall be generated by the pseudo random number generator inside TA.

6.2.3 Transmission of late entry synchronisation

After the first second, the maximum stealing rate of HIGH importance (see subclause 6.3) half-slots should not exceed an average of one per second. Half-slots marked as containing MEDIUM or LOW importance or marked as NO_VALID_DATA may be stolen more frequently to enhance the reliability of the synchronisation mechanism. However no more than four half-slots may be stolen per second. Insertion of synchronisation frames should not be regular, for example to make jamming more difficult.

Frame stealing by the encryption unit for late entry synchronisation shall not exceed the following average rates:

HIGH importance half slots: 1 per second;

MEDIUM importance half slots: 2 per second;

LOW importance half slots: 4 per second;

NO_VALID_DATA half slots: 4 per second.

An example of the "sync-control" process is given in subclause 6.6.

The SV's that are transmitted on behalf of late entry synchronisation are generated by the pseudo random number generator inside TA.

6.2.4 Reception of synchronisation

If the first half-slot of the transmission is received without detected errors (this should be indicated by the half-slot condition (HSC) indication) the receiver shall directly load the received SV into the receiver's KSG TA. The receiver should be in synchronisation and shall start generating SV's in the same way as the transmitter does. However if the first half-slot of the transmission cannot be decoded either due to channel errors or some other reason, the receiver will be unable to judge whether the second half slot is also stolen and containing synchronisation information, or if it contains user traffic. As it will be out of synchronisation, it should decode the second half slot as if it were synchronisation information, and if it decodes correctly, it should use the received value to initialise the receiver's key steam generator.

The chance of loss of synchronisation in less favourable radio conditions during reception should be reduced by a flywheel mechanism. In this flywheel mechanism the receiver should use locally generated SV's if an SV is not received correctly. When the receiver determines that an SV is received it should check whether this is correct or not (indicated by HSC). If the SV is correctly received this should be loaded into the pseudo random number generator of the receiver's KSG TA. If the SV is not received correctly the locally generated SV should be used. After a fixed number (N) of successive SV's are missed the receiver should be considered out of sync.

A state diagram of the receiver "sync-detect" process is given in figure 18.

6.2.5 Stolen frame format

The format of a stolen frame (half-slot) shall be as defined in Table 14:

Table 14: Stolen frame format (half-slot)

Information element	Element length	Element type	Reference	Remark
Half-slot stolen by encryption unit (HSSE)	1	1		
Stolen half-slot identifier (SHSI)	1	1		
Signalling data block	119	1		

Information elements coding:

Half-Slot Stolen by Encryption unit (HSSE): Indicates whether the frame is stolen by the encryption unit (1) or not (0).

Stolen Half-Slot Identifier (SHSI): Identifies the meaning of the signalling data in the stolen half-slot: encryption synchronisation frame (SF) (0) or other signalling data (1).

In case of an SF the signalling data block shall contain the following parameters:

- algorithm number (AN): 4 bits;
- key number (KN): 16 bits;
- replay offset select (ROS): 2 bits;
- synchronisation value (SV): 96 bits.

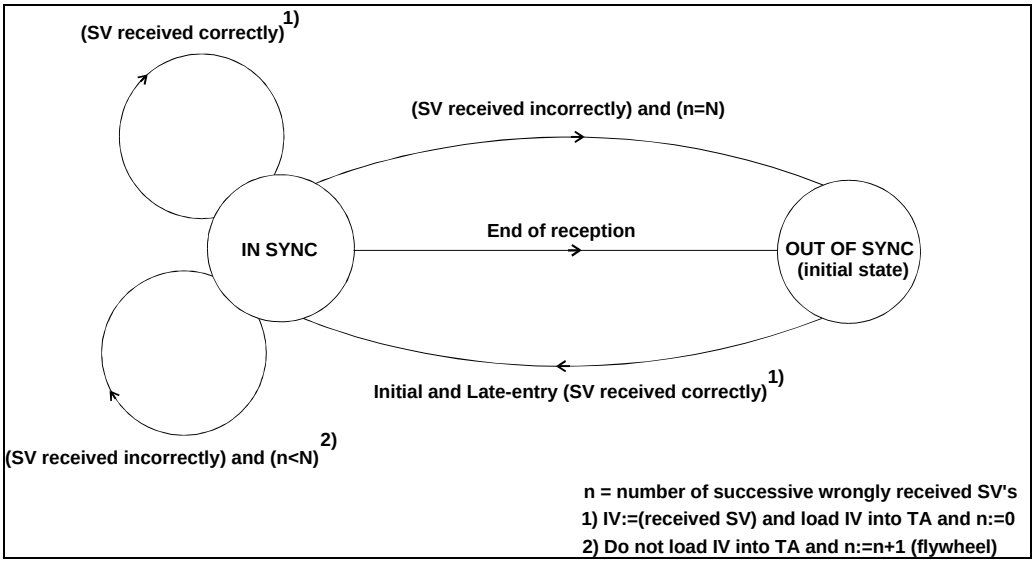


Figure 18: State diagram of the "sync-detect" process in the receiver

6.3 Location of security components in the functional architecture

This subclause describes the location of the encryption unit in the U-plane. Subclause 6.3.1 describes the codec-MAC interface. Subclause 6.3.2 gives the service primitives at the interface.

6.3.1 Codec-MAC interface

Figure 19 shows an SDL block diagram of the codec-MAC interface in the MS. The diagram shows the various blocks linked by channels carrying the indicated messages. The messages (defined in figure 20) carry data in the form of parameters. Skeleton type-definitions of the parameters are given in figure 21 and figure 22. The following features should be noted.

General:

1) The channel labelled “TMD-SAP” is the Service Access Point to the MAC layer.

2) A single encryption unit may be used for up to four concurrent calls, each connected to a different speech codec or data terminal with its own physical address, called a “u_device” in this ETS. A MS initiating a call can associate the call with an originating u_device, but there is no means of directing a call to a particular u_device in the destination MS. The receiving MS shall use local intelligence to identify the most appropriate u_device.

3) The message “TMD_REPORT_indication” shall have the following parameters:

   - “u_device” shall be the physical address of the source/destination of half-slots (e.g. codec or data port.) This enables parallel operation to take place by multiple units using different time-slots in the same radio unit. Up to 16 u_devices may be addressed;

   - “report”: The following values of the “report” parameter have been specified to enable the MAC to pass control information to the U-plane:

     - CALL_INFO: Shall provide information to the U-plane apparatus linking information concerning a call to the physical address of the u_device (e.g. a codec or data port) which is to participate in the call. Encryption on/off control, indication of the type of channel (TCH_type), slot-rate, the Call-ID and the end-user address identifier shall be provided by the CALL_INFO report messages from the MAC;

     - START_TX shall indicate that the MAC is ready to receive TMD_UNITDATA_request messages from the U-plane. Information concerning which half-slot is being provided for the first frame of data shall be given in the parameter “half_slot_number”;

     - STOP_TX Shall indicate that the U-plane should cease sending TMD_UNITDATA_request messages to the MAC from the specified u_device;

     - HALF_SLOT_STOLEN. Shall be sent when the MAC or encryption unit has stolen a half-slot sent from the U-plane data. The parameter “half_slot_number” shall be used to indicate which half-slot was stolen, so that the recipient can determine if the message refers to the half-slot just sent, or the one preceding it.

   - “TCH_type” shall only be relevant with report value “CALL_INFO”. It may have the following values:

     - TCH_S Speech call;

     - TCH_24 Circuit mode data at 2,4 kbit/s;

     - TCH_48 Circuit mode data at 4,8 kbit/s;

     - TCH_72 Circuit mode data at 7,2 kbit/s;

   - “slot-rate” shall only be valid with the report value “CALL_INFO”. It may take the values 1-4, depending upon the number of slots per frame allocated to the call;

   - “half_slot_number”. With report “START_TX” it shall indicate whether the next half-slot available for the U-plane to transmit is a first or second half-slot. With report “HALF_SLOT_STOLEN” it shall indicate whether a first or second half-slot was stolen. It is not used in other reports;

- "encryption_switch" shall only be valid with report value "CALL_INFO". It may have the values "ON" and "OFF", indicating whether encryption and decryption should be enabled or disabled;

- "call_id" shall only be valid with report value "CALL_INFO". It shall be used by the encryption unit for replay protection.

4) Message "TMD_UNITDATA_indication" shall be used to convey received half-slots from the MAC to the U-plane. It shall have the following parameters:

   - u_device: Destination address of data;

   - half_slot_number: FIRST_HALF_SLOT or SECOND_HALF_SLOT;

   - half_slot: This contains the half-slot data;

   - stolen_indication: NOT_STOLEN, C_STOLEN or U_STOLEN; where C_STOLEN indicates that the half slot has been stolen by the C-plane and the contents cannot be used by the U-plane. The C_STOLEN message permits the synchronisation to be maintained. U_STOLEN indicates data for use by the U-plane;

   - half_slot_condition: GOOD, BAD, or NULL; a half-slot should be marked by the MAC as NULL if the associated training sequence was not decoded, and as BAD if the training sequence was decoded but the CRC was faulty.

5) Message "TMD_UNITDATA_request" shall be the means of conveying half-slots from the U-plane to the MAC. It shall have the following parameters:

   - u_device: destination address of data;

   - half_slot_number: FIRST_HALF_SLOT or SECOND_HALF_SLOT;

   - half_slot: This contains the half-slot data;

   - stolen_indication: NOT_STOLEN or U_STOLEN;

   - importance: NO_VALID_DATA, LOW, MEDIUM, HIGH.

6) To support conformance testing it should be possible to disable signalling and encryption, to insert test input data and to read test output data.

7) Encryption state may only be changed during a call whilst no party in the call is transmitting. It may only be changed just prior to transmission by the party that is about to transmit.

Transmitter:

An example of message flow in transmit mode is given in subclause 6.7.

1) In the encrypted mode, speech data shall be passed from the speech coder to "end_switch" in the message "TMD_UNITDATA_request". End_switch shall pass the message to the encryption unit, which shall encrypt the speech data and then shall pass the message to the MAC via "sap" , which shall act as a route for messages to and from the MAC. This ensures a single interface between the MAC and speech coder/encryption unit and should permit the encryption unit to be removed.

2) The speech application may implement end-to-end signalling (U-plane signalling) in addition to that needed for encryption. One example maybe the control of comfort noise generation. To support this the "stolen_indication" shall also be passed from speech coder to encryption unit. If encryption is turned on, such signalling data shall be encrypted. However, certain data in the half-slot (HSSE and SHSI) shall be visible to the receiving U-plane before decryption, so the encryption algorithm for stolen half-slots shall not act on these bits.

3) The MAC should indicate to the speech application when a stealing has occurred, using a TMD_REPORT_indication. If possible, this signal should be sent before the data for the next half-slot is transferred from the speech coder, as the coder may be able to compensate for the stolen

speech frame. The speech coder can discover if the stealing indication refers to the most recent half-slot by checking the value of the “half_slot_number” parameter. The stealing report shall not be sent if it is two or more half-slots late.

4) The following messages from speech coder to encryption unit shall be valid:

- Two half-slots of speech;
- One half-slots of U-plane signalling and one half-slot of speech;
- Two half-slots of U-plane signalling.

5) Since the encryption unit shall not normally be allowed to steal U-plane signalling, the following messages from the encryption unit shall be valid:

- Two half-slots of encrypted speech;
- One half-slot of U-plane signalling and one half-slot of encrypted speech;
- Two half-slots of U-plane signalling;
- One half-slot of encryption sync and one half-slot of encrypted speech;
- Two half-slots of encryption sync.

Receiver:

An example of message flow in receive mode is given in subclause 6.7.

1) Messages from the MAC shall be routed by the block labelled “sap” to either the end_switch or the encryption unit, depending whether encryption is switched off or on. The end_switch shall forward the data to the codec.

2) The channel decoder shall be informed if a half-slot has been stolen.

3) The Half-Slot Stolen indicator at the output of the channel decoder shall be associated with each half-slot.

4) The decryption unit shall be capable of recognising signalling data intended for the speech or other applications and decrypting it.

5) The decryption unit shall not decrypt half-slots marked with a NULL half_slot_condition. It shall decrypt BAD half-slots as the end-user may be able to obtain some information from them.

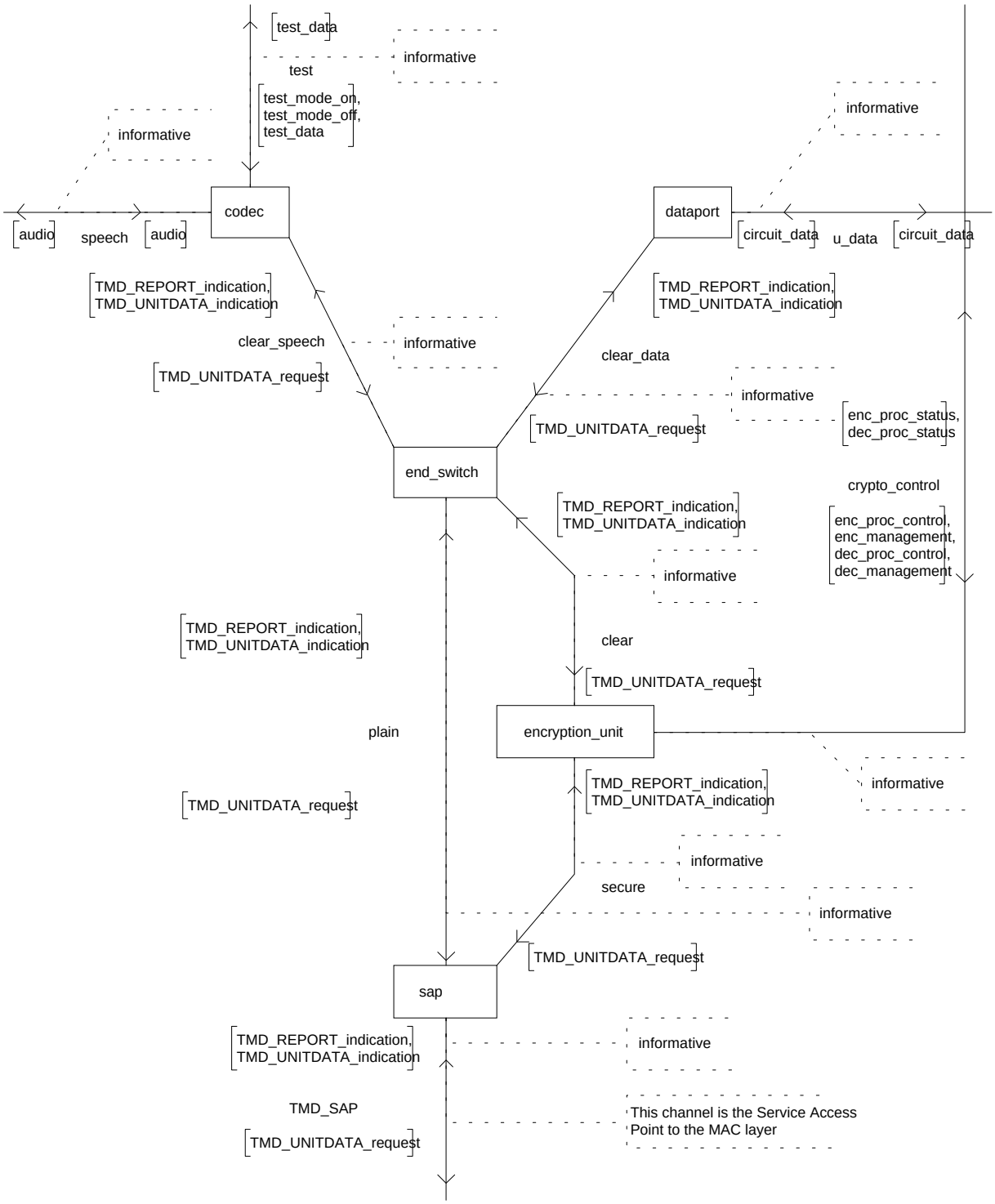


Figure 19: Model of U_plane components

```
/*** SIGNAL DEFINITIONS ***/

SIGNAL

  TMD_REPORT_indication(
        report, half_slot_number, TCH_type, slot_rate, encryption_switch, call_id, u_device),
  TMD_UNITDATA_indication(
       half_slot_number, half_slot, stolen_indication, half_slot_condition, u_device),
  TMD_UNITDATA_request(
        half_slot_number, half_slot, stolen_indication, importance, u_device),
  test_mode_on,
  test_mode_off,
  test_data(test_vectors),
  audio(samples),
  circuit_data(data_block),
  enc_proc_status(cipher_report, cipher_id, u_device),
  enc_proc_control(
        algorithm_number, key_number, replay_offset_select, replay_offset,
        init_value, u_device),
  dec_proc_control(
        algorithm_number, key_number, replay_offset_select, replay_offset,
       init_value, u_device),
 enc_management(cipher_key, algorithm_number, key_command, key_number, u_device),
 dec_proc_status(cipher_report, cipher_id, u_device),
 dec_management(cipher_key, algorithm_number, key_command, key_number, u_device);
```

Figure 20: Signal definitions at U-plane interfaces

```
/*** DATA TYPE DEFINITIONS ***/

NEWTYPE report
   LITERALS CALL_INFO, START_TX, STOP_TX, HALF_SLOT_STOLEN;
ENDNEWTYPE report;

NEWTYPE TCH_type
   LITERALS TCH_S, TCH_24, TCH_48, TCH_72;
ENDNEWTYPE TCH_type;

SYNTYPE slot_rate = natural
   CONSTANTS 1:4
ENDSYNTYPE slot_rate;

NEWTYPE encryption_switch
   LITERALS ON, OFF
ENDNEWTYPE encryption_switch;

NEWTYPE call_id
   CONSTANTS 0:16383      /* 14 bits */
ENDNEWTYPE call_id;

NEWTYPE half_slot_number
   LITERALS FIRST_HALF_SLOT, SECOND_HALF_SLOT
ENDNEWTYPE half_slot_number;

NEWTYPE half_slot
   LITERALS dummy              /* 216 bits */
ENDNEWTYPE half_slot;

NEWTYPE stolen_indication
    LITERALS NOT_STOLEN, C_STOLEN, U_STOLEN
ENDNEWTYPE stolen_indication;

NEWTYPE half_slot_condition
   LITERALS GOOD, BAD, NULL
ENDNEWTYPE half_slot_condition;

NEWTYPE u_device
   LITERALS CODEC_1, DATAPORT_1  /*  up to 16 device addresses */
ENDNEWTYPE u_device;

NEWTYPE test_vectors
   LITERALS dummy              /*  1920 bits */
ENDNEWTYPE test_vectors;
```

Figure 21: Type definitions for U-plane interfaces

```
/*** DATA TYPE DEFINITIONS ***/

NEWTYPE samples
  LITERALS dummy         /* 1920 bits */
ENDNEWTYPE samples;

NEWTYPE data_block
  LITERALS dummy        /* size determined by external data protocol */
ENDNEWTYPE data_block;

NEWTYPE importance
  LITERALS NO_VALID_DATA, LOW, MEDIUM, HIGH
ENDNEWTYPE importance;

SYNTYPE key_number = natural
  CONSTANTS 0:65535      /* 16 bits */
ENDSYNTYPE key_number;

SYNTYPE algorithm_number = natural
  CONSTANTS 0:15         /* 4 bits */
ENDSYNTYPE algorithm_number;

SYNTYPE replay_offset_select = natural
  CONSTANTS  1:4
ENDSYNTYPE replay_offset_select;

NEWTYPE replay_offset
  LITERALS dummy        /* 128 bits */
ENDNEWTYPE replay_offset;

NEWTYPE cipher_key
  LITERALS dummy        /* 128 bits */
ENDNEWTYPE cipher_key;

NEWTYPE init_value
  LITERALS dummy        /* 96 bits */
ENDNEWTYPE init_value;

NEWTYPE key_command
  LITERALS KEY_DATA, KEY_DELETE, KEY_SELECT, CRYPTO_IDENTIFY_DEMAND
ENDNEWTYPE key_command;

NEWTYPE cipher_report
  LITERALS KEY_ACKNOWLEDGE, DELETE_ACKNOWLEGE, CRYPTO_IDENTIFY_RESPONSE,
        IN_SYNC, OUT_OF_SYNC
ENDNEWTYPE cipher_report;

NEWTYPE cipher_id
  ARRAY(cipher_id_index, character)
  LITERALS dummy
ENDNEWTYPE cipher_id;

SYNTYPE cipher_id_index = Integer
  CONSTANTS 0:19
ENDSYNTYPE cipher_id_index;
```

Figure 22: Type definitions (concluded)

6.3.2 Primitive description

To permit the interface of a variety of end-to-end encryption algorithms/systems, the interface between the algorithm and its host has been specified at a logical level. The main features of the interface are:

- support of up to 16 algorithm types/modes within the encryption unit;
- separate addressing, keying and control for each algorithm type;
- provision for self synchronising and synchronous ciphers;
- support for routing of data (voice) to four different end-points;
- possibility for separation of clear, secure and control data at the interface (implementation dependent).

Three bi-directional channels shall be implemented:

crypto_control: the channel shall carry signals from the host to configure the encryption unit for use. It shall return status indications;

clear: the channel shall carry the clear signal TMD_UNITDATA_request from the host to the encryption unit and shall return the clear TMD_UNITDATA_indication and the status TMD_REPORT_indication to the host;

secure: the channel shall carry the secure signal TMD_UNITDATA_request from the host and shall return the secure signal TMD_UNITDATA_indication and the status TMD_REPORT_indication to the host;

The service primitives at the interface shall be as follows:

enc_management:

crypto_control channel.

The signal shall permit an algorithm to be initialised with encryption keys. It shall also permit implementation specific key commands to be passed. Such commands could be used to modify the algorithm mode of operation or assist in end-to-end re-keying.

Table 15: enc_management; crypto_control channel

Parameter	Request parameter type
u_device	M
algorithm_number	M
key_number	O
key_command	O
cipher_key	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

enc_proc_control:

crypto_control channel.

The signal shall permit an algorithm to be configured for use in encryption of data coming from a user device.

The key to be used for encryption shall be identified by key_number. The selection of a mode of replay protection shall be performed by replay_offset_select. If replay protection is enabled, the optional field replay_offset shall be included.

Table 16: enc_proc_control; crypto_control channel

Parameter	Request parameter type
u_device	M
algorithm_number	M
key_number	O
replay_offset_select	M
replay_offset	C
init_value	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

dec_management:

crypto_control channel.

The signal shall permit an algorithm to be initialised with the keys for decryption. It shall also permit implementation specific key commands to be passed. Such commands may be used to modify the algorithm mode of operation or assist in end-to-end re-keying.

Table 17: dec_management; crypto_control channel

Parameter	Request parameter type
u_device	M
algorithm_number	M
key_number	O
key_command	O
cipher_key	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

dec_proc_control:

crypto_control channel.

The signal shall permit an algorithm to be configured for use in decryption of data to a user device.

The key to be used for decryption shall be identified by key_number. The selection of a mode of replay protection shall be performed by replay_offset_select. If replay protection is enabled, the optional field replay_offset shall be included.

Table 18: dec_proc_control; crypto_control channel

Parameter	Request parameter type
u_device	M
algorithm_number	M
key_number	O
replay_offset_select	M
replay_offset	C
init_value	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

TMD_UNITDATA primitive:

clear and secure channels.

TMD_UNITDATA_request shall contain the traffic data (half_slot) and associated addressing and control data. When passed to the encryption unit, half_slot shall contain unencrypted voice or user data. When sent from the encryption unit to TMD_SAP (and encryption is enabled for this u_device),half_slot shall be encrypted voice or user data or it shall have been replaced by synchronisation or control information associated with the end-to-end encryption process.

If a half-slot has been stolen by the encryption unit, it shall be indicated to the MAC by the appropriate setting of stolen_indication. If the half-slot contains information essential to the decryption units, importance should be set to indicate high, irrespective of its previous value.

TMD_UNITDATA_indication shall contain the traffic data (half_slot) and associated addressing and control data. If a half-slot has not been received, it shall still be necessary to pass a NULL half-slot to the encryption unit in order to maintain correct bit timing for synchronous stream ciphers. When sent from the encryption unit to a u_device, half_slot shall be unencrypted voice, user data or NULL.

Table 19: TMD_UNITDATA primitive; clear and secure channels

Parameter	Request parameter type	Indication parameter type
u_device	M	M
half_slot_number	M	M
stolen_indication	M	M
half_slot_condition	-	M
importance	M	-
half_slot	M	M

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

TMD_REPORT primitive:

clear and secure channels.

Table 20: TMD_REPORT primitive; clear and secure channels

Parameter	Indication parameter type
u_device	M
half_slot_number	M
report	O
encryption_switch	O
call_id	O
TCH_type	O
slot_rate	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

enc_process_status:

crypto_control channel.

This signal shall permit an algorithm to pass status and control information regarding the encryption process to the host.

Table 21: enc_process_status; crypto_control channel

Parameter	Indication parameter type
u_device	M
cipher_report	O
cipher_id	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

dec_process_status:

crypto_control channelThis signal shall permit an algorithm to pass status and control information regarding the decryption process to the host.

Table 22: dec_process_status; crypto_control channel

Parameter	Indication parameter type
u_device	M
cipher_report	O
cipher_id	O

KEY: M: Mandatory; C: Conditional; O: Optional; -: Not used

The parameters used in the primitives should be as follows:

report =

1) CALL_INFO;
2) START_TX;
3) STOP_TX;
4) HALF_SLOT_STOLEN.

TCH_type =

1) TCH_S;
2) TCH_24;
3) TCH_48;
4) TCH_72.

slot_rate =

1) 1;
2) 2;
3) 3;
4) 4.

encryption_Swpwitch =

1) ON;
2) OFF.

call_id =

1) 0;
2) 1;
...
n) 16383.

half_slot_number =

1) FIRST_HALF_SLOT;
2) SECOND_HALF_SLOT.

half_slot =

1) 0;
2) 1;
...
n) 2^{216} - 1.

stolen_indication =

1) NOT_STOLEN;
2) C_STOLEN;
3) U_STOLEN.

half_slot_condition =

1) GOOD;
2) BAD;
3) NULL.

u_device =

1) 0;
2) 1;
...
n) 15.

importance =

1) NO_VALID_DATA;
2) LOW;
3) MEDIUM;
4) HIGH.

key_number =

1) 0;
2) 1;
...
n) 65535.

algorithm_number =

1) 0;
2) 1;
...
n) 15.

replay_offset_select =

1) SELECTION_1;
2) SELECTION_2;
3) SELECTION_3;
4) SELECTION_4.

replay_offset =

1) 0;
2) 1;
...
n) 2^{128} - 1.

cipher_key =

1) 0;
2) 1;
...
n) 2^{128} - 1.

init_value =

1) 0;
2) 1;
...
n) 2^{96} - 1.

key_command =

1) KEY_DATA;
2) KEY_DELETE;
3) KEY_SELECT;
4) CRYPTO_IDENTIFY_DEMAND.

cipher_report =

1) KEY_ACKNOWLEDGE;
2) DELETE_ACKNOWLEDGE;
3) CRYPTO_IDENTIFY_RESPONSE;
4) IN_SYNC;
5) OUT_OF_SYNC.

cipher_id =

1) String of 20 ASCII characters-1;
2) String of 20 ASCII characters-2;
...
n) String of 20 ASCII characters-N.

6.4 Definition of boundary conditions

TA: shall be used to generate the key stream segment KSS.

Input 1: Bit string DK of length 128;
Input 2: Bit string IV of length 96;

Output 1: Bit stream KSS;
Output 2: Bit string SV of length 96.

The algorithm should be designed such that the key stream is pseudo random with a very long period and that the chance of using the same part in the key stream during this period in case IV and/or DK is changed, shall be negligible. This also should be the case if a number of DK's with known differences are applied. SV shall be the content of a register used in the pseudo random number generator.

6.5 Dimensioning of security values

Algorithm Number (AN): 4 bits.

Cipher Key (CK): 128 bits.

Derived Key (DK): 128 bits.

Initialisation Value (IV): 96 bits.

Key Number (KN0): 16 bits.

Replay Offset Select (ROS): 2 bits.

Synchronisation Value (SV): 96 bits.

6.6 Example of “sync-control” process

Figure 23 gives an example for determining the points of time of transmitting a new SV by the “sync-control” process. Transmission of a new SV shall be forced after a period of 1 second after the last transmission of an SV. More SV’s may be transmitted to improve reliability of synchronisation and to allow fast late entry. This can be done only under the condition that the HSI of the half-slots coming from the codec is equal to MEDIUM, LOW or NO_VALID_DATA. However no more than 4 SV’s may be transmitted per second.

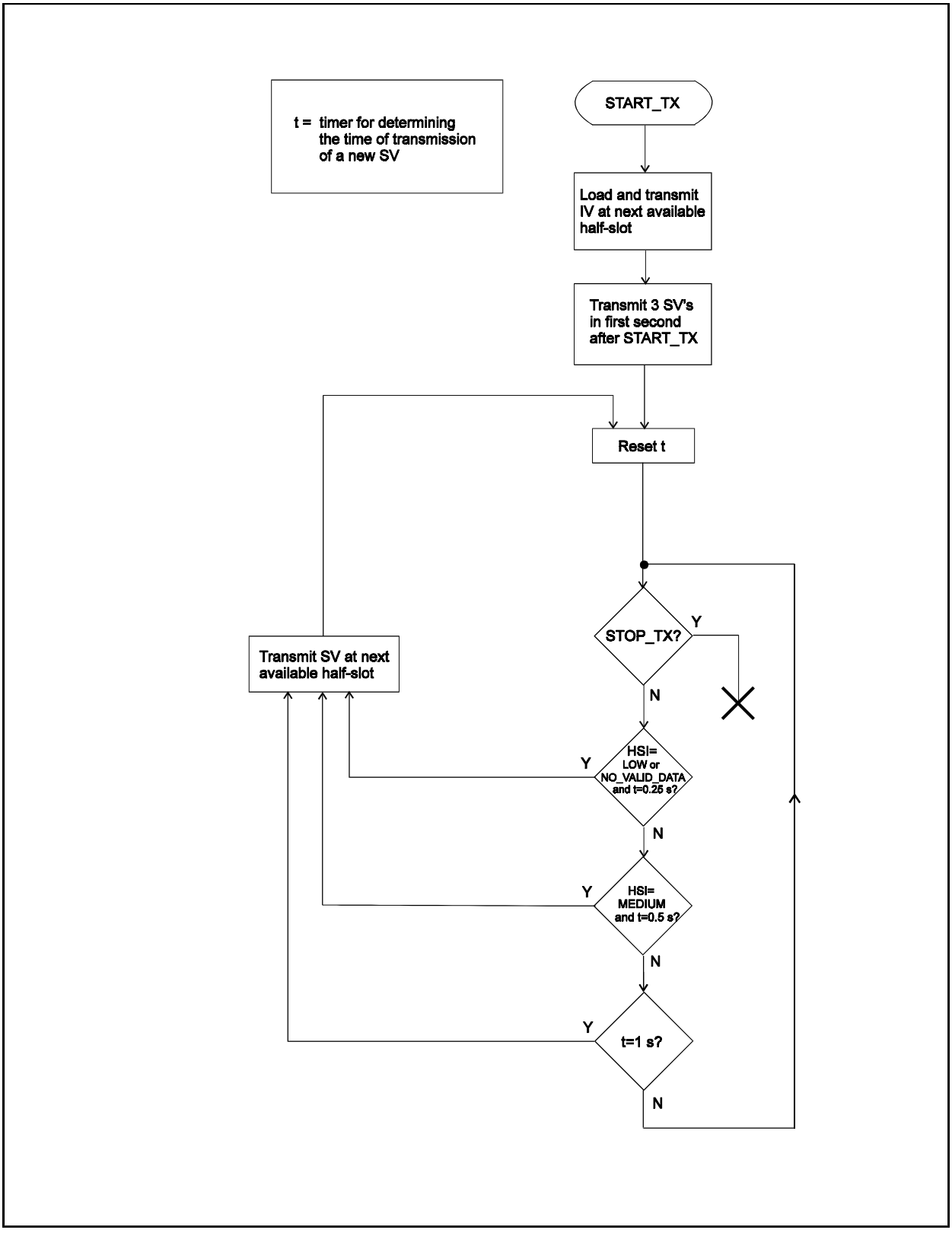


Figure 23: Flow chart of the transmitter “sync-control” process

6.7 Message flow examples

1) A TMD_REPORT_indication from the MAC (CALL_INFO) shall inform the U-plane components that future half-slots transmitted to and from the codec with address CODEC_1 should be encrypted speech (TCH_S).

2) A TMD_REPORT_indication from the MAC (START_TX) shall invite the codec (CODEC_1) to start sending half-slots.

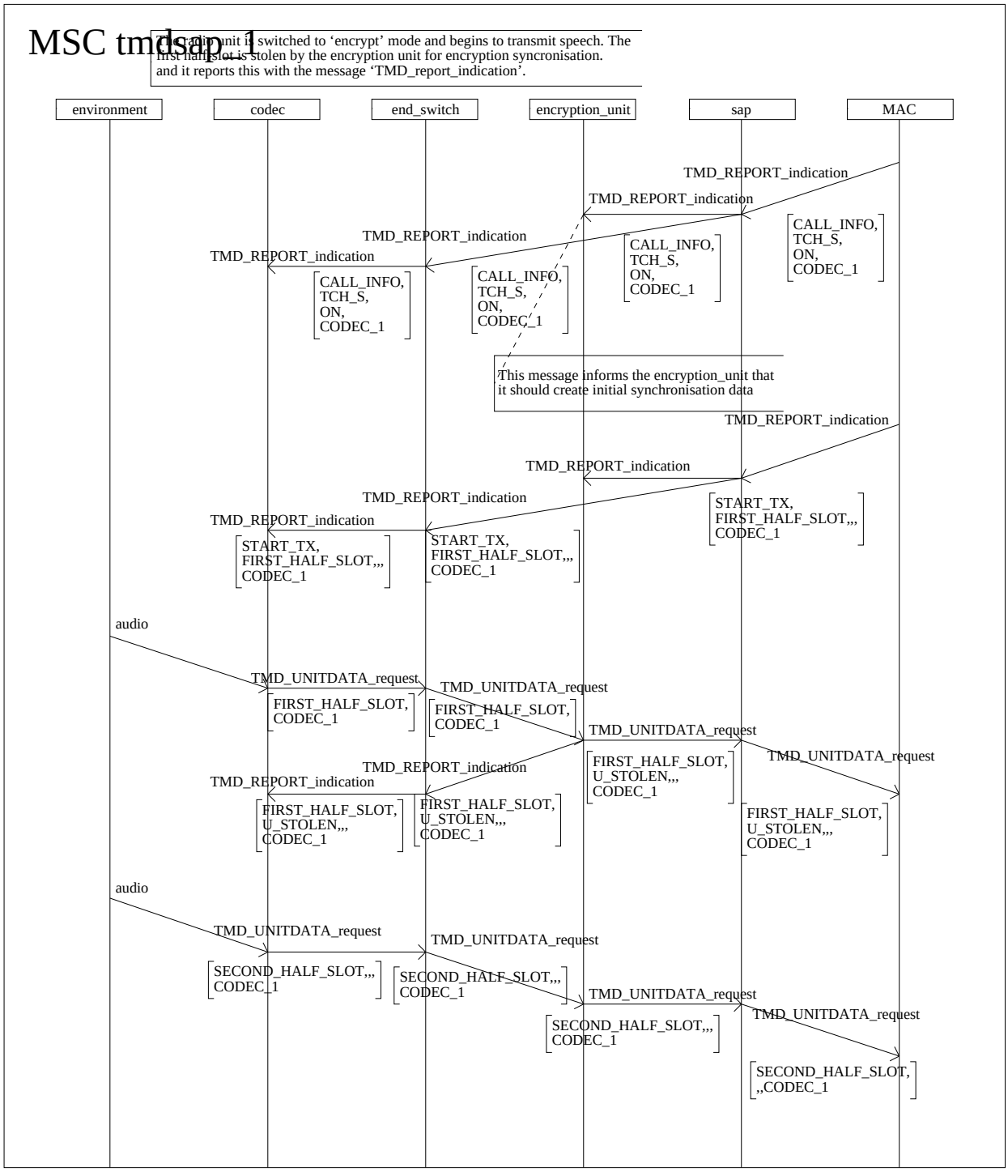


Figure 24: Transmission of initial synchronisation

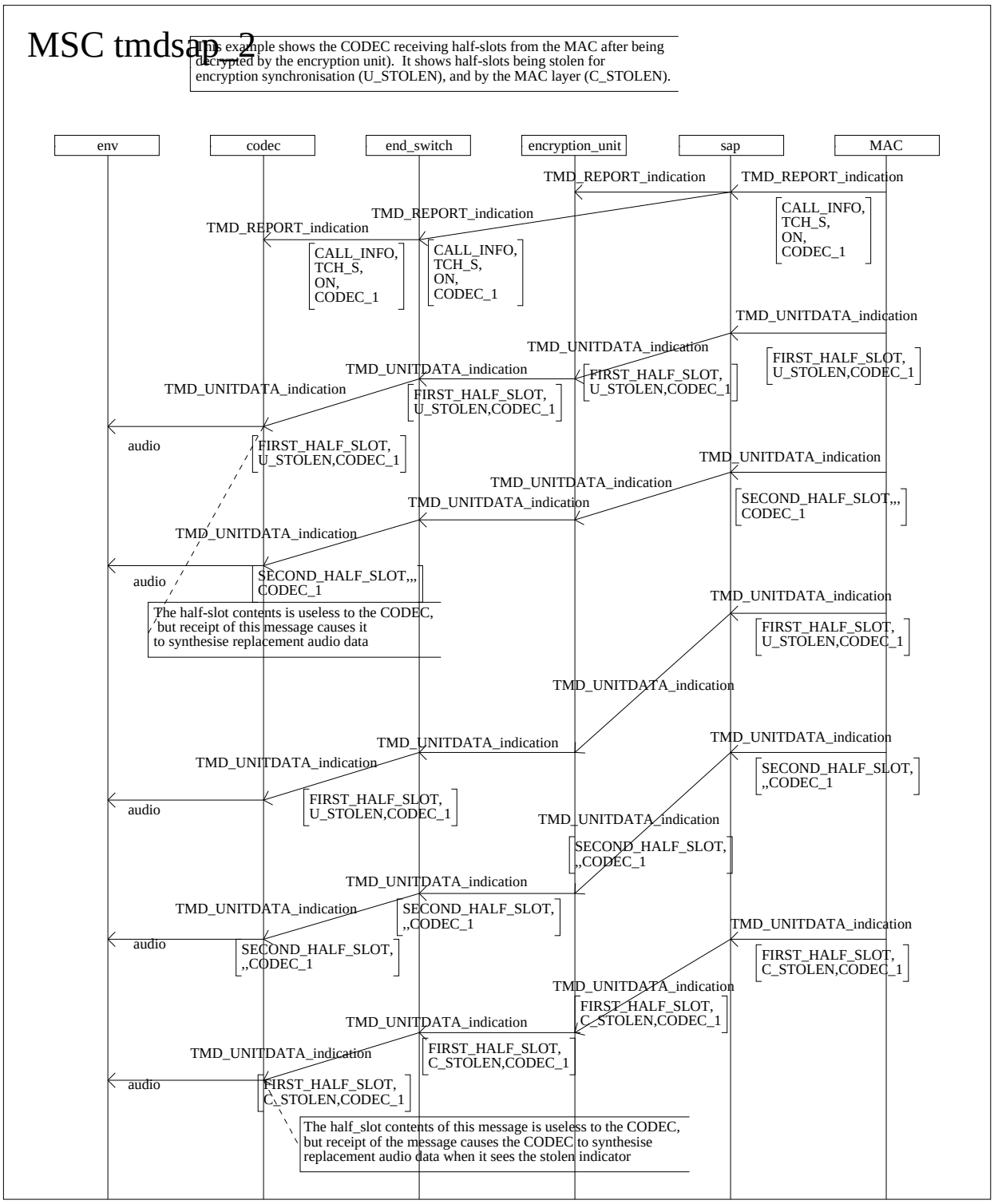


Figure 25: Reception of stolen half-slots

History

Document history			
September 1995	Public Enquiry	PE 92:	1995-09-25 to 1996-01-19
May 1996	Converted into Adobe Acrobat Portable Document Format (PDF)		