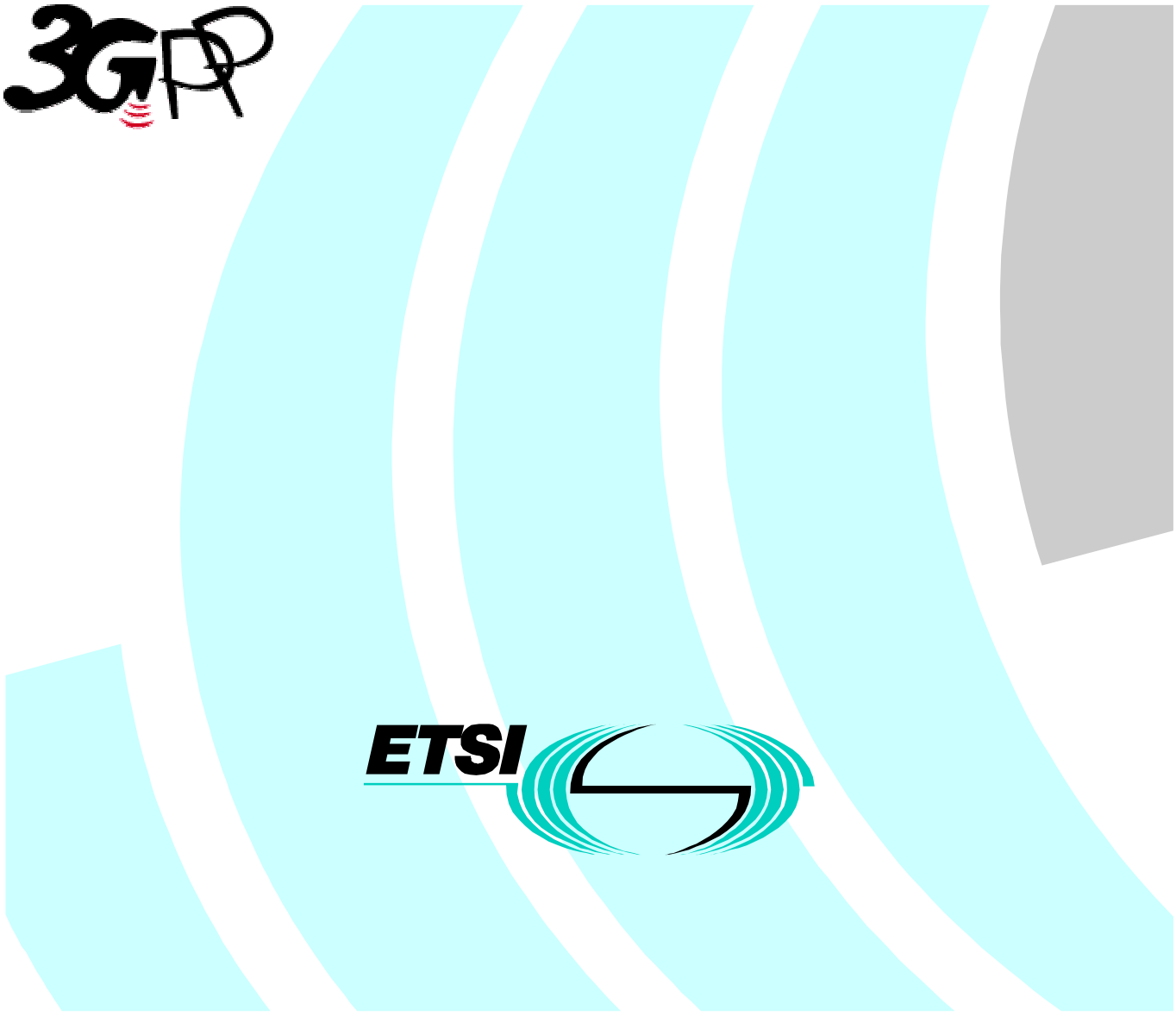
ETSI TS 133 105 V3.6.0 (2000-12)

Technical Specification

Universal Mobile Telecommunications System (UMTS);
3G Security;
Cryptographic Algorithm Requirements
(3GPP TS 33.105 version 3.6.0 Release 1999)



Reference
RTS/TSGS-0333105UR4

Keywords
UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by the ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under www.etsi.org/key .

Contents

Foreword

This Technical Specification has been produced by the 3GPP.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version 3.y.z

where:

3 the first digit:

3 Indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification.

1 Scope

This specification constitutes a requirements specification for the security functions which may be used to provide the network access security features defined in [1].

The specification covers the intended use of the functions, the technical requirements on the functions and the requirements as regards standardization.

For those functions that require standardization, it also covers the intended use of the algorithm specification, the requirements on test data, and quality assurance requirements on both the algorithm and its documentation.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

[1] 3G TS 33.102: "3rd Generation Partnership Project (3GPP); Technical Specification Group (TSG) SA; 3G Security; Security Architecture".

[2] Wassenaar Arrangement, December 1998.

[3] ISO/IEC 9797: "Information technology - Security techniques - Data integrity mechanism using a cryptographic check function employing a block cipher algorithm".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following definitions apply:

Confidentiality: The property that information is not made available or disclosed to unauthorised individuals, entities or processes.

Data integrity: The property that data has not been altered in an unauthorised manner.

Data origin authentication: The corroboration that the source of data received is as claimed.

Entity authentication: The provision of assurance of the claimed identity of an entity.

Key freshness: A key is fresh if it can be guaranteed to be new, as opposed to an old key being reused through actions of either an adversary or authorised party.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

$\|\|$	Concatenation
$\oplus$	Exclusive or

f0	random challenge generating function
f1	network authentication function
f1*	the re-synchronisation message authentication function;
f2	user authentication function
f3	cipher key derivation function
f4	integrity key derivation function
f5	anonymity key derivation function for normal operation
f5*	anonymity key derivation function for re-synchronisation
f8	UMTS encryption algorithm
f9	UMTS integrity algorithm

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3GPP	3rd Generation Partnership Project
AK	Anonymity key
AuC	Authentication Centre
AUTN	Authentication token
COUNT-C	Time variant parameter for synchronisation of ciphering
COUNT-I	Time variant parameter for synchronisation of data integrity
CK	Cipher key
IK	Integrity key
IMSI	International Mobile Subscriber Identity
IPR	Intellectual Property Right
MAC	Medium access control (sublayer of Layer 2 in RAN)
MAC	Message authentication code
MAC-A	MAC used for authentication and key agreement
MAC-I	MAC used for data integrity of signalling messages
PDU	Protocol data unit
RAND	Random challenge
RES	User response
RLC	Radio link control (sublayer of Layer 2 in RAN)
RNC	Radio network controller
SDU	Signalling data unit
SQN	Sequence number
UE	User equipment
USIM	User Services Identity Module
XMAC-A	Expected MAC used for authentication and key agreement
XMAC-I	Expected MAC used for data integrity of signalling messages
XRES	Expected user response

4 General algorithm requirements

4.1 Resilience

The functions should be designed with a view to its continued use for a period of at least 20 years. Successful attacks with a work load significantly less than exhaustive key search through the effective key space should be impossible.

The designers of above functions should design algorithms to a strength that reflects the above qualitative requirements.

4.2 World-wide availability and use

Legal restrictions on the use or export of equipment containing cryptographic functions may prevent the use of such equipment in certain countries.

It is the intention that UE and USIMs which embody such algorithms should be free from restrictions on export or use, in order to allow the free circulation of 3G terminals. Network equipment, including RNC and AuC, may be expected to come under more stringent restrictions. It is the intention is that RNC and AuC which embody such algorithms should be exportable under the conditions of the Wassenaar Arrangement [2].

5 Functional algorithm requirements

5.1 Authentication and key agreement

5.1.1 Overview

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

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

5.1.1.1 Generation of quintets in the AuC

To generate a quintet the HLR/AuC:

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

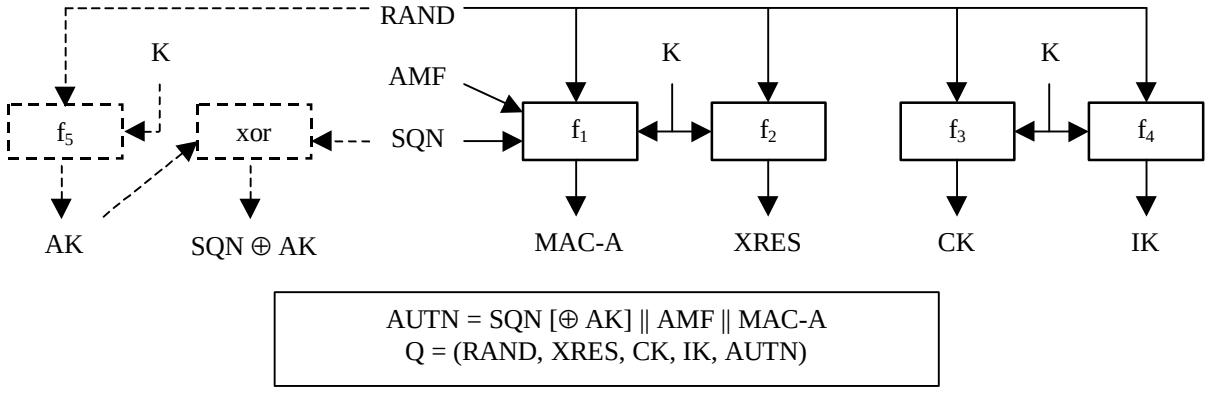


Figure 1: Generation of quintets in the AuC

5.1.1.2 Authentication and key derivation in the USIM

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

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

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

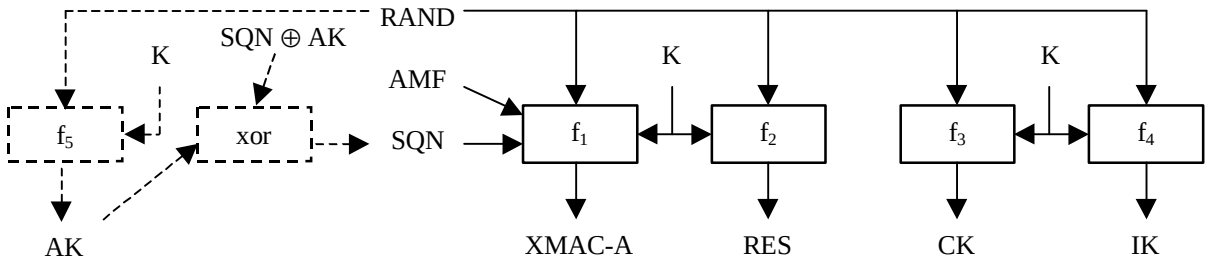


Figure 2: Authentication and key derivation in the USIM

5.1.1.3 Generation of re-synchronisation token in the USIM

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

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

b) If SQN_{MS} is to be concealed with an anonymity key AK, the USIM computes AK = $f5^*_K$(RAND), and the concealed counter value is then computed as SQN_{MS} ⊕ AK.

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

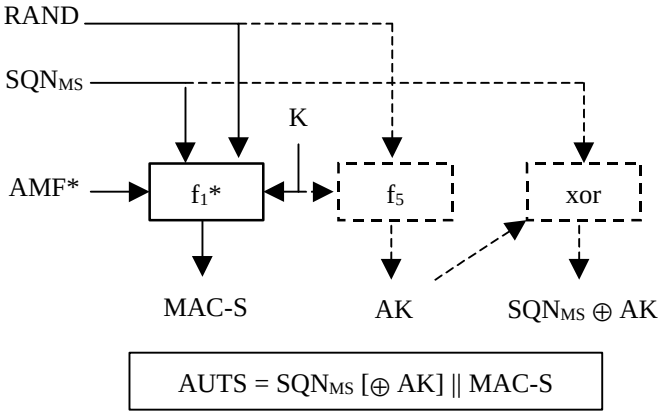


Figure 3: Generation of re-synchronisation token in the USIM

5.1.1.4 Re-synchronisation in the HLR/AuC

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

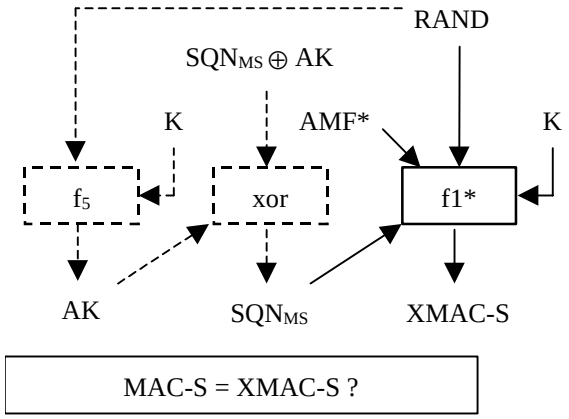


Figure 4: Re-synchronisation in the HLR/AuC

a) If SQN_{MS} is concealed with an anonymity key AK, the HLR/AuC computes AK = $f5^*_K$(RAND)and retrieves the unconcealed counter value as SQN_{MS} = (SQN_{MS} ⊕ AK) xor AK.

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

5.1.2 Use

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

5.1.3 Allocation

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

5.1.4 Extent of standardisation

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

5.1.5 Implementation and operational considerations

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

5.1.6 Type of algorithm

5.1.6.1 f0

f0: the random challenge generating function

f0: (internal state) → RAND

f0 should be (pseudo) random number generating function.

5.1.6.2 f1

f1: the network authentication function

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

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

5.1.6.3 f1*

f1*: the re-synchronisation message authentication function

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

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

5.1.6.4 f2

f2: the user authentication function

f2: (K; RAND) → RES (or XRES)

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

5.1.6.5 f3

f3: the cipher key derivation function

f3: (K; RAND) → CK

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

5.1.6.6 f4

f4: the integrity key derivation function

f4: (K; RAND) → IK

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

5.1.6.7 f5

f5: the anonymity key derivation function for normal operation

f5: (K; RAND) → AK

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

The use of f5 is optional.

5.1.6.8 f5*

f5*: the anonymity key derivation function for re-synchronisation

f5*: (K; RAND) → AK

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

The use of f5* is optional.

5.1.7 Interface

5.1.7.1 K

K: the subscriber authentication key

K[0], K[1], …, K[127]

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

5.1.7.2 RAND

RAND: the random challenge

RAND[0], RAND[1], …, RAND[127]

The length of RAND is 128 bits.

5.1.7.3 SQN

SQN: the sequence number

SQN[0], SQN[1], …, SQN[47]

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

5.1.7.4 AMF

AMF: the authentication management field

AMF[0], AMF[1], …, AMF[15]

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

5.1.7.6 MAC-A (equivalent for XMAC-A)

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

MAC-A[0], MAC-A[1], …, MAC-A[63]

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

5.1.7.7 MAC-S (equivalent for XMAC-S)

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

MAC-S[0], MAC-S[1], …, MAC-S[63]

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

5.1.7.8 RES (or XRES)

RES: the user response

RES[0], RES[1], …, RES[n-1]

The length n of RES and XRES is at most 128 bits and at least 32 bits. RES and XRES constitute to entity authentication of the user to the network.

5.1.7.9 CK

CK: the cipher key

CK[0], CK[1], …, CK[127]

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

5.1.7.10 IK

IK: the integrity key

IK[0], IK[1], …, IK[127]

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

5.1.7.11 AK

AK: the anonymity key

AK[0], AK[1], …, AK[47]

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

5.2 Data confidentiality

5.2.1 Overview

The mechanism for data confidentiality of user data and signalling data that is described in 6.6 of [1] requires the following cryptographic function:

f8 UMTS encryption algorithm.

Figure 1 illustrates the use of f8 to encrypt plaintext by applying a keystream using a bitwise XOR operation. The plaintext may be recovered by generating the same keystream using the same input parameters and applying it to the ciphertext using a bitwise XOR operation.

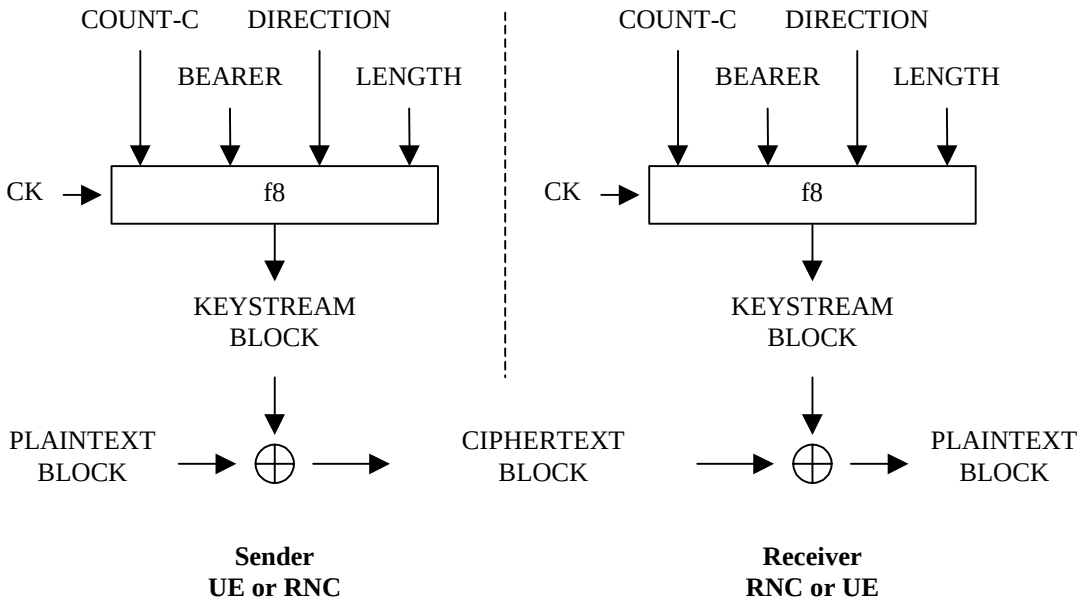


Figure 1: Ciphering user and signalling data transmitted over the radio access link

The input parameters to the algorithm are the Cipher Key (CK), a time dependent input (COUNT-C), the bearer identity (BEARER), the direction of transmission (DIRECTION) and the length of the keystream required (LENGTH). Based on these input parameters the algorithm generates the output keystream block (KEYSTREAM) which is used to encrypt the input plaintext block (PLAINTEXT) to produce the output ciphertext block (CIPHERTEXT).

The input parameter LENGTH shall affect only the length of the KEYSTREAM BLOCK, not the actual bits in it.

5.2.2 Use

The function f8 shall only be used to protect the confidentiality of user data and signalling data sent over the radio access link between UE and RNC.

5.2.3 Allocation

The function f8 is allocated to the UE and the RNC.

Encryption will be applied in the Medium Access Control (MAC) sublayer and in the Radio Link Control (RLC) sublayer of the data link layer (Layer 2).

5.2.4 Extent of standardisation

The function f8 shall be fully standardized.

5.2.5 Implementation and operational considerations

The algorithm should be designed to accommodate a range of implementation options including hardware and software implementations. For hardware implementations, it should be possible to implement one instance of the algorithm using less than 10,000 gates (working assumption).

A wide range of UE with different bearer capabilities is expected, so the encryption throughput requirements on the algorithm will vary depending on the implementation. However, based on the likely maximum user traffic data rates, it must be possible to implement the algorithm to achieve an encryption speed in the order of 2Mbit/s on the downlink and on the uplink.

1. RLC-transparent mode:
   - New keystream block required every physical layer frame (10ms)
   - Maximum number of bits per physical layer frame of 20000 bits
   - Minimum number of bits per physical layer frame of 1 bit
   - Granularity of 1 bit on all possible intermediate values.
2. For UM RLC mode:
   - New keystream block required per UMD PDU
   - Maximum number of bits in UMD PDU is 5000 bits
   - Minimum number of bits in UMD PDU is 16 bits
   - Granularity of 8 bit on all possible intermediate values.
3. For AM RLC mode:
   - New keystream block required per AMD PDU
   - Maximum number of bits in AMD PDU is 5000 bits
   - Minimum number of bits in AMD PDU is 24 bits
   - Granularity of 8 bit on all possible intermediate values.

The encryption throughput requirements should be met based on clock speeds upwards of 20MHz (typical clock speeds are expected to be much greater than this).

5.2.6 Type of algorithm

The function f8 should be a symmetric synchronous stream cipher.

5.2.7 Interfaces to the algorithm

5.2.7.1 CK

CK: the cipher key

CK[0], CK[1], …, CK[127]

The length of CK is 128 bits. In case the effective key length k is smaller than 128 bits, the most significant bits of CK shall carry the effective key information, whereas the remaining, least significant bits shall repeat the effective key information:

$CK[n] = CK[n \bmod k]$, for all n, such that $k \leq n < 128$.

5.2.7.2 COUNT-C

COUNT-C: the cipher sequence number.

COUNT-C[0], COUNT-C[1], …, COUNT-C[31]

The length of the COUNT-C parameter is 32 bits.

Sychronisation of the keystream is based on the use of a physical layer (Layer 1) frame counter combined with a hyperframe counter introduced to avoid re-use of the keystream. This allows the keystream to be synchronised every 10ms physical layer frame. The exact structure of the COUNT-C is specified in TS 33.102.

5.2.7.3 BEARER

BEARER: the radio bearer identifier.

BEARER[0], BEARER[1], …, BEARER[4]

The length of BEARER is 5 bits.

The same cipher key may be used for different radio bearers simultaneously associated with a single user which are multiplexed onto a single 10ms physical layer frame. To avoid using the same keystream to encrypt more than one bearer, the algorithm shall generate the keystream based on the identity of the radio bearer.

5.2.7.4 DIRECTION

DIRECTION: the direction of transmission of the bearer to be encrypted.

DIRECTION[0]

The length of DIRECTION is 1 bit.

The same cipher key may be used for uplink and downlink channels simultaneously associated with a UE, which are multiplexed onto a single 10ms physical layer frame. To avoid using the same keystream to encrypt both uplink and downlink transmissions, the algorithm shall generate the keystream based on the direction of transmission.

The value of the DIRECTION is 0 for messages from UE to RNC and 1 for messages from RNC to UE.

An explicit direction value is required in preference to splitting the keystream segment into uplink and downlink portions to allow for asymmetric bearer services.

5.2.7.5 LENGTH

LENGTH: the required length of keystream.

LENGTH[0], LENGTH[1], …, LENGTH[15]

The length of LENGTH is 16 bits.

For a given bearer and transmission direction the length of the plaintext block that is transmitted during a single physical layer frame may vary. The algorithm shall generate a keystream block of variable length based on the value of the length parameter.

The input parameter LENGTH shall affect only the length of the KEYSTREAM BLOCK, not the actual bits in it.

The maximum RLC PDU / MAC SDU size is 5000 bits. The range of values of the length parameter will depend not only on the RLC PDU / MAC SDU size but also the number of RLC PDUs / MAC SDUs which may be sent in a single physical layer 10ms frame for a given bearer and transmission direction.

Not all values between the maximum and minimum values shall be required but it is expected that the ability to produce length values of whole numbers of octets between a minimum and a maximum value will be required.

5.2.7.6 KEYSTREAM

KEYSTREAM: the output keystream.

KS [0], KS [1], …, KS [LENGTH-1]

The length of a keystream block equals the value of the input parameter LENGTH.

5.2.7.7 PLAINTEXT

PLAINTEXT: the plaintext.

PT[0], PT[1], …, PT[LENGTH-1]

The length of a keystream block equals the value of the input parameter LENGTH.

This plaintext block consists of the payload of the particular RLC PDUs / MAC SDUs to be encrypted for a given bearer and transmission direction. It may consist of user traffic or signalling data:

- For RLC UM mode, the plaintext block is the UMD PDU excluding the first octet, i.e. excluding the RLC UM PDU header (see TS 25.322 [19]).
- For RLC AM mode, the plaintext block is the AMD PDU excluding the two first octets, i.e. excluding the RLC AM PDU header (see TS 25.322 [19]).
- For RLC TM on DCH, the plaintext block consists of all the MAC SDUs containing data for one and the same radio bearer and sent in one Transmission Time Interval. In this case, the CFN part of COUNT-C for the plaintext block is the CFN for the first radio frame of the Transmission Time Interval containing the plaintext block. (see TS 25.321 [18]).

5.2.7.8 CIPHERTEXT

CIPHERTEXT: the ciphertext.

CT[0], CT[1], …, CT[LENGTH-1]

The length of a keystream block equals the value of the input parameter LENGTH.

5.3 Data integrity

5.3.1 Overview

The mechanism for data integrity of signalling data that is described in 6.6 of [1] requires the following cryptographic function:

f9 UMTS integrity algorithm.

Figure 3 illustrates the use of the function f9 to derive a MAC-I from a signalling message.

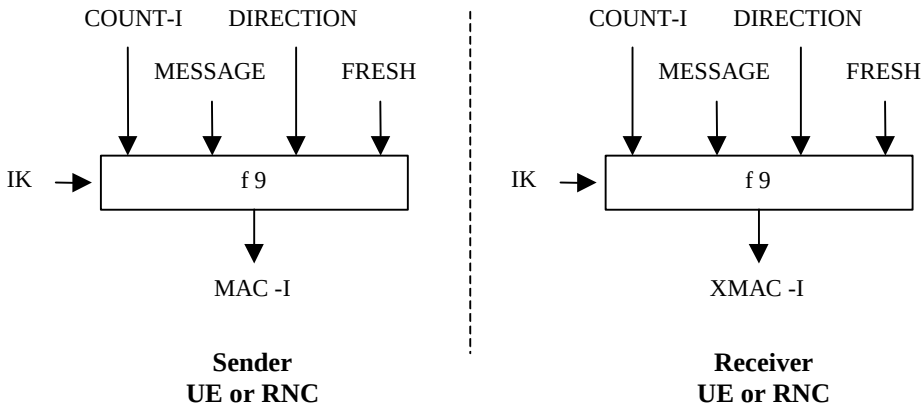


Figure 2: Derivation of MAC-I (or XMAC-I) on a signalling message

The input parameters to the algorithm are the Integrity Key (IK), a time dependent input (COUNT-I), a random value generated by the network side (FRESH), the direction bit (DIRECTION) and the signalling data (MESSAGE). Based on these input parameters the user computes with the function f9 the message authentication code for data integrity (MAC-I) which is appended to the message when sent over the radio access link. The receiver computes XMAC-I on the messages received in the same way as the sender computed MAC-I on the message sent.

5.3.2 Use

The MAC function f9 shall be used to authenticate the data integrity and data origin of signalling data transmitted between UE and RNC.

5.3.3 Allocation

The MAC function f9 is allocated to the UE and the RNC.

Integrity protection shall be applied at the RRC layer.

5.3.4 Extent of standardisation

The function f9 is fully standardized.

5.3.5 Implementation and operational considerations

The algorithm should be designed to accommodate a range of implementation options including hardware and software implementations.

5.3.6 Type of algorithm

The function f9 shall be a MAC function.

5.3.7 Interface

5.3.7.1 IK

IK: the integrity key

IK[0], IK[1], …, IK[127]

The length of IK is 128 bits.

5.3.7.2 COUNT-I

COUNT-I: a frame dependent input.

COUNT-I[0], COUNT-I[1], …, COUNT-I[31]

The length of COUNT-I is 32 bits.

The input parameter COUNT-I protects against replay during a connection. It is a value incremented by one for each integrity protected message. COUNT-I consists of two parts: the HYPERFRAME NUMBER (HFN) as the most significant part and a RRC Sequence Number as the least significant part. The initial value of the hyperframe number is sent by the user to the network at connection set-up. The user stores the greatest used hyperframe number from the previous connection and increments it by one. In this way the user is assured that no COUNT-I value is re-used (by the network) with the same integrity key.

5.3.7.3 FRESH

FRESH: a random number generated by the RNC.

FRESH[0], FRESH[1], …, FRESH[31]

The length of FRESH is 32 bits.

The same integrity key may be used for several consecutive connections. This FRESH value is an input to the algorithm in order to assure the network side that the user is not replaying old MAC-Is.

5.3.7.4 MESSAGE

MESSAGE: the signalling data.

MESSAGE[0], MESSAGE[1], …, MESSAGE[X19-1]

The maximum length of MESSAGE is X19.

5.3.7.5 DIRECTION

DIRECTION: the direction of transmission of signalling messages (user to network or network to users).

DIRECTION[0]

The length of DIRECTION is 1 bit.

The same integrity key may be used for uplink and downlink channels simultaneously associated with a UE.

The value of the DIRECTION is 0 for messages from UE to RNC and 1 for messages from RNC to UE.

5.3.7.6 MAC-I (and equivalently XMAC-I)

MAC-I: the message authentication code for data integrity authentication

MAC-I[0], MAC-I[1], …, MAC-I[31]

The length of MAC-I is 32 bits.

6 Use of the algorithm specifications

The purpose of this clause is to address ownership of the algorithm specification, to define which types of organisation are entitled to use the algorithm specification, and to outline how and under what conditions such organisations may obtain the specification.

6.1 Ownership

For those functions which require to be fully standardized, all copyright on the algorithm and test data specifications shall be owned jointly by the 3GPP partner organisations.

6.2 Design authority

The design authority for the algorithms that require standardisation shall be ETSI SAGE. It is expected that the project team assembled by SAGE will draw on appropriate expertise within the 3GPP partner organisations in addition to its normal resource pool.

6.3 Users of the specification

For those functions which require to be fully standardized, the algorithm specification shall be published as a 3GPP specification. It will be used by those who need the algorithm specification to build equipments or components which embody the algorithm.

6.4 Licensing

For those functions which require to be fully standardized, the use of the algorithm shall be subject to a license agreement which restricts the use of the algorithm as described in 5.3.2 and 5.4.2.

Users of the algorithm, and users of the algorithm specification, shall be required to sign the licence agreement. Appropriate licence agreements shall be drawn up by the 3GPP partner organisations.

Licences shall be royalty free. In addition, the licence agreement shall require users of the specification not to attempt to patent the algorithm or otherwise register an Intellectual Property Right (IPR) relating to the algorithm or its use.

6.5 Management of the specification

For those functions which require to be fully standardized, the algorithm specifications shall be published as a 3GPP specification. The algorithms will thus be open for public evaluation. It is recognised that this will leave the algorithms open to public criticism during the commercial operation of the system. The process of responding to public criticism will need to be handled carefully by an appropriate 3GPP body.

7 Algorithm specification and test data requirements

For those functions that require standardization, the design authority should provide four separate deliverables: a specification of the algorithm, a set of design conformance test data, a set of algorithm input/output test data and a design and evaluation report. Requirements on the specification and test data deliverables are given in this clause, those on the design and evaluation report in 9.3.

7.1 Specification of the algorithm

An unambiguous specification of the algorithm needs to be provided which is suitable for use by implementers of the algorithm.

The specification shall include an annex which provides simulation code for the algorithm written in ANSI C. The specification may also include an annex containing illustrations of functional elements of the algorithm.

7.2 Implementors test data

The implementors test data is required to assist implementors of the algorithm in their realisation of the algorithm specification.

This set of test data, as well as including algorithm input and output data, shall include details of the internal state of the algorithm at various stages in its execution. Sufficient detail shall be provided to enable implementators to readily identity the likely location of any errors in their implementation.

Final validation of the implementation shall be performed using the design conformance test data (see subclause 7.3).

7.3 Design conformance test data

Design conformance test data is required to allow implementers of the algorithm to validate their implementations, and manufacturers to validate embodiments of the algorithm (e.g. in ASICs or FPGAs).

The test data set shall be presented as input/output test data, allowing the realisation to be tested as a 'black box'. (i.e. the test data shall consist solely of data passed across the interfaces to the algorithm.)

The design conformance test data shall be designed to give a high degree of confidence in the correctness of any implementation of the algorithm. The set of test data shall ensure that all elements of the algorithm are fully exercised.

7.4 Format and handling of deliverables

The specification of the algorithm shall be produced on paper, and published as a 3GPP specification.

The algorithm input/output test data shall be produced on paper and on magnetic disc, and published by 3GPP. The document and disc shall be provided to 3GPP partner organisations.

8 Quality assurance requirements

This clause advises the design authority on measures needed to provide users of the algorithm with confidence that it is fit for purpose, and users of the algorithm specification and test data assurance that appropriate quality control has been exercised in their production.

The measures shall be recorded by the design authority in a design and evaluation report which shall be published as a 3GPP specification.

8.1 Quality assurance for the algorithm

Prior to its release to 3GPP, the algorithm needs to be approved as meeting the functional requirements specified in clause 7 by all members of the design authority.

8.2 Quality assurance for the specification and test data

Prior to delivery of the algorithm specification, two independent simulations of the algorithm needs to be made using the specification, and confirmed against test data designed to allow verification of significant points in the execution of the algorithm.

Design conformance and algorithm input/output test data needs to be generated using a simulation of the algorithm produced from the specification and confirmed as above. The simulation used to produce this test data needs to be identified in the test data deliverables and retained by the design authority.

8.3 Design and evaluation report

The design and evaluation report is intended to provide evidence to potential users of the algorithm, specification and test data that appropriate and adequate quality control has been applied to their production. The report shall explain the following:

- the algorithm and test data design criteria;
- the algorithm evaluation criteria;
- the methodology used to design and evaluate the algorithm;
- the extent of the mathematical analysis and statistical testing applied to the algorithm;
- the principal conclusions of the algorithm evaluation;
- the quality control applied to the production of the algorithm specification and test data.

The report shall confirm that all members of the design authority have approved the algorithm, specification and test data.

The report shall contain key conclusions from the commissioned closed evaluation of the algorithm.

9 Summary of the design authority deliverables

For those cryptographic functions that require standardisation, the design authority shall deliver:

- Specification of the algorithm;
- Implementors test data;
- Design conformance test data;
- Design and evaluation report.

All these documents shall be delivered to 3GPP for subsequent publication.

Annex A (informative): Void

Void.

Annex B (informative): Change history

Change history					
TSG SA #	**Version**	**CR**	**Tdoc SA**	**New Version**	**Subject/Comment**
SA#04				3.0.0	Approved at SA#04 and placed under TSG SA Change Control
SA#05	3.0.0	001	SP-99418	3.1.0	Resources for cryptographic algorithms in the USIM
SA#05	3.0.0	002	SP-99418	3.1.0	MAC used for data integrity of signaling messages
SA#05	3.0.0	003	SP-99418	3.1.0	Cipher keystream block length
SA#06	3.1.0	004	SP-99587	3.2.0	Time variant parameter for synchronisation of ciphering
SA#06	3.1.0	005	SP-99587	3.2.0	Direction bit in f9
SA#07	3.2.0	006	SP-000048	3.3.0	Authentication and key agreement
SA#07	3.2.0	007	SP-000048	3.3.0	Enhanced user confidentiality (Some conflict with CR008. CR008 and 33.102 used as basis for terminology)
SA#07	3.2.0	008	SP-000075	3.3.0	Refinement of EUIC for consistency with 33.102
SA#07	3.2.0	009	SP-000048	3.3.0	Ciphering
SA#07	3.2.0	010	SP-000048	3.3.0	Data integrity
SA#08	3.3.0	011	SP-000271	3.4.0	Clarification of BEARER and DIRECTION parameters
SA#09	3.4.0	012	SP-000445	3.5.0	Calculation of AK in re-synchronisation
SA#09	3.4.0	013	SP-000444	3.5.0	Deletion of eUIC
SA#09	3.4.0	014	SP-000445	3.5.0	Anonymity key computation during re-synchronisation
SA#10	3.5.0	015	SP-000627	3.6.0	Layer 2 related corrections

History

Document history		
V3.2.0	January 2000	Publication
V3.3.0	March 2000	Publication
V3.4.0	July 2000	Publication
V3.5.0	October 2000	Publication
V3.6.0	December 2000	Publication