Technical Specification Group Services and System Aspects ***TSG SA#4(99)296***
Meeting #4, Miami, USA, 21-23 June 1999

TSG SA WG3 #4, London, 16-18 June 1999 **S3-99201**

Subject: 3G TS 33.105 V1.0.0

Source: TSG SA WG3

Document for: Information and Approval

3G TS 33.105 V1.0.0 (1999-06)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
3G Security;
Cryptographic Algorithm Requirements
(3G TS 33.105 version 1.0.0)

Reference

DTS/TSGS-0333120U

Keywords

Security, Authentication and key agreement, Ciphering Algorithm,
Message Authentication Code Algorithm

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Subject: 3G TS 33.105 V1.0.0 1
Foreword 6
1 Scope 6
2 References 6
3 Definitions, symbols and abbreviations 6
3.1 Definitions 6
3.2 Symbols 6
3.3 Abbreviations 7
4 General algorithm requirements 7
4.1 Resilience 7
4.2 World-wide availability and use 8
5 Functional algorithm requirements 8
5.1 Authentication and key agreement 8
5.1.1 Overview 8
5.1.2 Use 10
5.1.3 Allocation 10
5.1.4 Extent of standardisation 10
5.1.5 Implementation and operational considerations 10
5.1.6 Type of algorithm 10
5.1.6.1 f0 10
5.1.6.2 f1 10
5.1.6.3 f1* 11
5.1.6.4 f2 11
5.1.6.5 f3 11
5.1.6.6 f4 11
5.1.6.7 f5 11
5.1.7 Interface 11
5.1.7.1 K 11
5.1.7.2 RAND 12
5.1.7.3 SQN 12
5.1.7.4 MODE 12
5.1.7.6 MAC-A (equivalent for XMAC-A) 12
5.1.7.7 MAC-S (equivalent for XMAC-S) 12
5.1.7.8 RES (or XRES) 12
5.1.7.9 CK 13
5.1.7.10 IK 13
5.1.7.11 AK 13
5.2 Data confidentiality 13
5.2.1 Overview 13
5.2.2 Use 14
5.2.3 Allocation 14
5.2.4 Extent of standardisation 14
5.2.5 Implementation and operational considerations 14
5.2.6 Type of algorithm 15
5.2.7 Interfaces to the algorithm 15
5.2.7.1 CK 15
5.2.7.2 COUNT 15
5.2.7.3 BEARER 15

5.2.7.4 DIRECTION ... 15
5.2.7.5 LENGTH ... 16
5.2.7.6 KEYSTREAM ... 16
5.2.7.7 PLAINTEXT ... 16
5.2.7.8 CIPHERTEXT ... 16
5.3 Data integrity ... 17
5.3.1 Overview ... 17
5.3.2 Use ... 17
5.3.3 Allocation ... 17
5.3.4 Extent of standardisation ... 17
5.3.5 Implementation and operational considerations ... 17
5.3.6 Type of algorithm ... 18
5.3.7 Interface ... 18
5.3.7.1 IK ... 18
5.3.7.2 COUNT ... 18
5.3.7.3 FRESH ... 18
5.3.7.4 MESSAGE ... 18
5.3.7.5 MAC-I (and equivalently XMAC-I) ... 18

6 Use of the algorithm specifications ... 19
6.1 Ownership ... 19
6.2 Design authority ... 19
6.3 Users of the specification ... 19
6.4 Licensing ... 19
6.5 Management of the specification ... 19

7 Algorithm specification and test data requirements ... 19
7.1 Specification of the algorithm ... 20
7.2 Implementors test data ... 20
7.3 Design conformance test data ... 20
7.4 Format and handling of deliverables ... 20

8 Quality assurance requirements ... 20
8.1 Quality assurance for the algorithm ... 20
8.2 Quality assurance for the specification and test data ... 21
8.3 Design and evaluation report ... 21

9 Summary of the design authority deliverables ... 21

Annex A (informative): User identity confidentiality ... 22
A.1 Overview ... 22
A.2 Use ... 22
A.3 Allocation ... 22
A.4 Extent of standardisation ... 22
A.5 Implementation and operational considerations ... 22
A.6 Type of algorithm ... 23
A.6.1 f6 ... 23
A.6.2 f7 ... 23
A.7 Interface ... 23
A.7.1 GK ... 23
A.7.2 SEQ_UIC ... 23
A.7.3 IMUI ... 23
A.7.4 EMUI ... 23

Annex B: Functions for alternative AKA protocol ... 24
B.1 Scope ... 24
B.2 Abbreviations ... 24
B.3 General algorithm requirements ... 24

B.4 Functional algorithm requirements 24
B.4.1 Overview 24
B.4.2 Use 26
B.4.3 Extend of standardisation 26
B.4.4 Allocation 26
B.4.5 Implementation and operational considerations 26
B.4.6 Type of algorithms 26
B.4.6.1 f0 26
B.4.6.2 f1 27
B.4.6.3 f2 27
B.4.6.4 f3 27
B.4.6.5 f4 27
B.4.6.6 f5 28
B.4.7 Interface 28
B.4.7.1 KT 28
B.4.7.2 RSn and RSu 28
B.4.7.3 RES1, XRES1, RES2, XRES2 28

Annex C: History 29

Annex D: Unspecified values 30

Foreword

TBA

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.

[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]	ISO9797.

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

⊕	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.
f6	user identity encryption function
f7	user identity decryption function
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
CK	Cipher key
EMUI	Encrypted Mobile User Identity
GK	User group key
IK	Integrity key
IMUI	International Mobile User 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
SEQ_UIC	Sequence for user identity confidentiality
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.

Figure 1 illustrates the use of the cryptographic functions f0—f5 at the AuC.

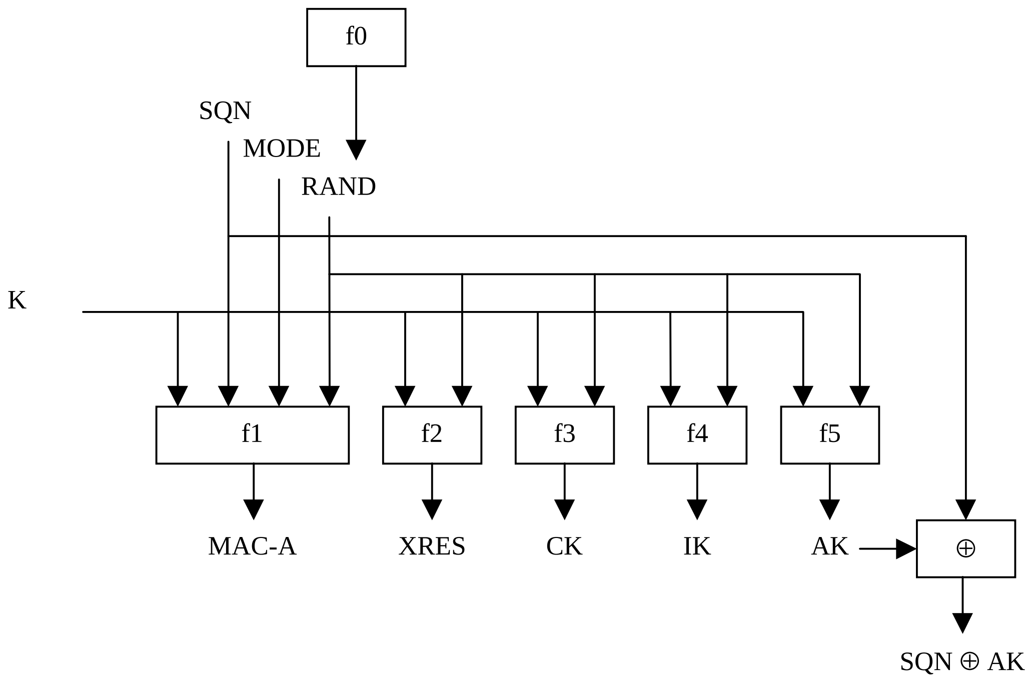


Figure 1: Functions for authentication and key agreement in the AuC

The input parameters to the algorithms are the long-term secret Key (K), the Sequence Number (SQN), a random challenge (RAND) generated by f0 and a mobility management mode indicator (MODE). Based on K and RAND f2—f5 compute an expected RESponse (XRES), a Cipher Key (CK), an Integrity Key (IK) and an Anonymity Key (AK) which is bitwise XOR-ed with SQN. Based on K, RAND, SQN and MODE, f1 derives a Message Authentication Code for network to user Authentication (MAC-A).

Figure 2 illustrates the use of function f1* in the AuC .

The input parameters to the algorithm f1* are the long-term secret Key (K), the Sequence Number (SQN), a random challenge (RAND) received by the AuC from the SN and a mobility management mode indicator (MODE). Based on these input parameters, an expected message authentication code XMAC-S on the synchronisation failure data is computed.

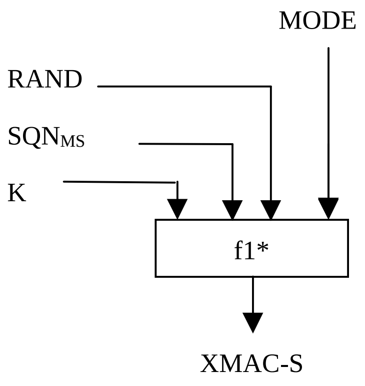


Figure 2: The re-synchronisation message authentication function in the AuC

Figure 3 illustrates the use of the cryptographic functions f1—f5 in the USIM.

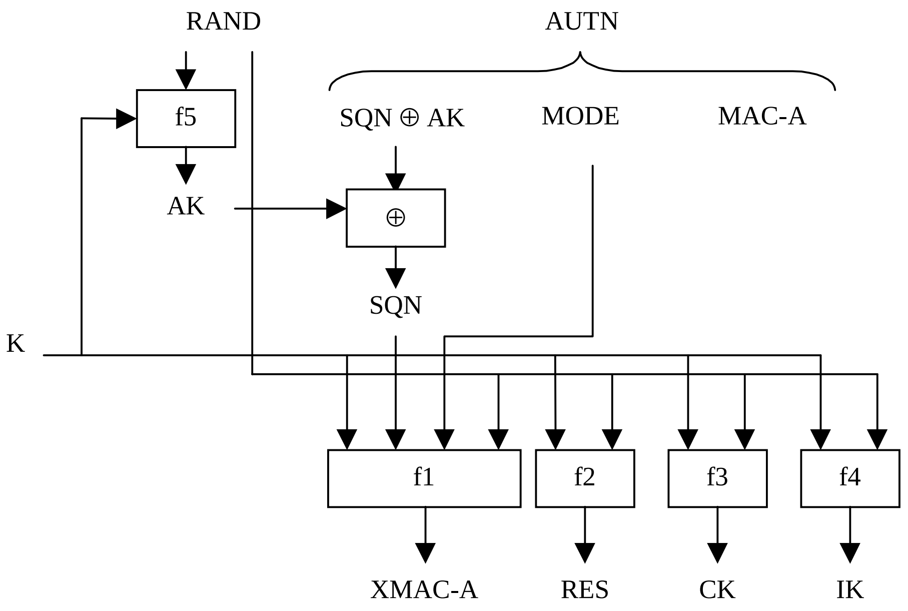


Figure 3: Functions for authentication and key agreement in the USIM

This time the Anonymity Key (AK) has to be computed in advance to retrieve SQN from SQN ⊕ AK. From there on, f1 is used to derive XMAC-A from K, RAND, SQN and MODE and f2—f4 are used to derive RES, CK and IK in the same way as the AuC derived MAC-A, XRES, CK and IK.

The use of the function in f1* in the USIM is strictly analogous to that in the AuC. A figure is therefore omitted. Based on the input parameters long-term secret Key (K), Sequence Number (SQN), random challenge (RAND) received by the USIM from the SN and mobility management mode indicator (MODE), a message authentication code MAC-S on synchronisation failure data is computed.

Note: An alternative mechanism for authentication and key agreement that requires a different set of cryptographic functions is included in Annex D of [1]. The requirements on the cryptographic functions required for that mechanism are described in Annex A of this specification.

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.

5.1.3 Allocation

The functions f1—f5 and f1*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 and f1*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 and f1* shall be designed so that they can be implemented on an IC card equipped with a X1-bit microprocessor running at X2 MHz with X3 kbits of memory and produce AK, MAC-A and RES in less than X4 ms, and CK and IK in less than X5 ms.

The functions f0—f5 and f1* shall be designed so that they can be implemented either in software or in hardware in the AuC on a X6-bit microprocessor running at X7 MHz and with X8 kbits of memory and produce MAC-A, XRES, CK, IK and AK in less than X9 ms.

5.1.6 Type of algorithm

5.1.6.1 f0

f0: the random challenge generating function

f0: (internal state) $\rightarrow$ RAND

f0 should be (pseudo) random number generating function.

5.1.6.2 f1

f1: the network authentication function

f1: (K; SQN, RAND, MODE) $\rightarrow$ 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, MODE and MAC-A (or XMAC-A).

5.1.6.3 f1*

f1*: the re-synchronisation message authentication function

f1*: (K; SQN, RAND, MODE) $\rightarrow$ 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, MODE and MAC-S (or XMAC-S).

5.1.6.4 f2

f2: the user authentication function

f2: (K; RAND) $\rightarrow$ 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) $\rightarrow$ 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) $\rightarrow$ 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

f5: (K; RAND) $\rightarrow$ AK

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

5.1.7 Interface

5.1.7.1 K

K: the user authentication key

K[0], K[1], …, K[127]

The length of K is 128 bits. The user 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[X10-1]

The length of SQN is between 32 and 64 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 MODE

MODE: the mobility management identifier

MODE[0]

The length of MODE is 1 bit. MODE identifies an instance of mobility management, when more than one mobility management is simultaneously active for a single user.

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 MODE. 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 MODE. 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[X21]

or XRES: the expected user response

XRES[0], XRES[1], …, XRES[X21]

The maximum length of RES and XRES is 128 bits and the minimum is 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[X10-1]

The length of AK is between 32 and 64 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 6.4 of [1] requires the following cryptographic function:

f8 UMTS encryption algorithm.

Figure 2 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.

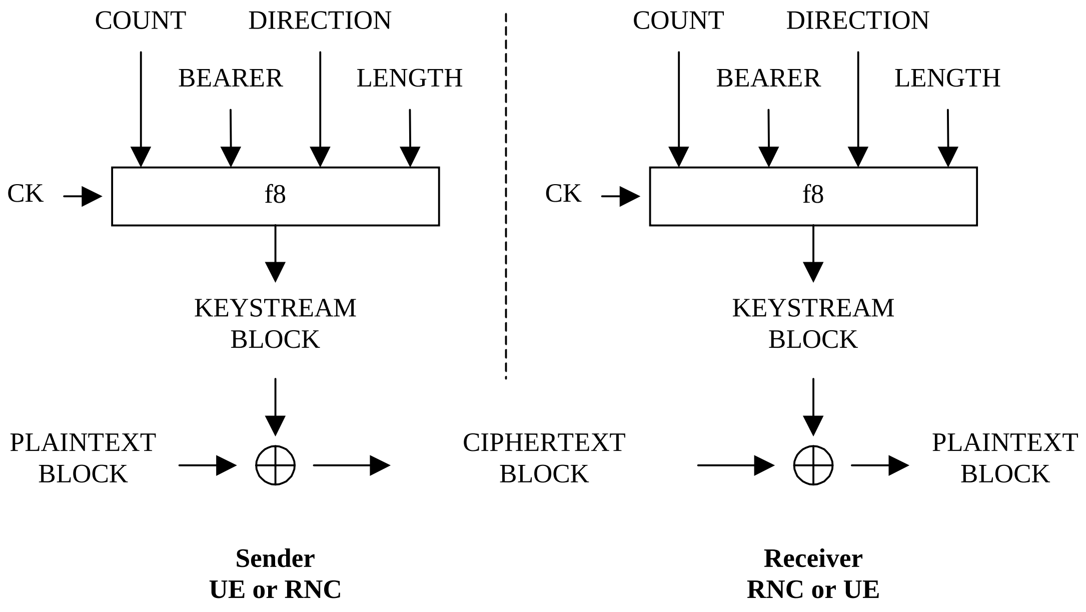


Figure 2: Ciphering user and signalling data transmitted over the radio access link

The input parameters to the algorithm are the Cipher Key (CK), a time dependent input (COUNT), 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). It is assumed that synchronisation of the keystream will be 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. It must be noted that these details are subject to change based on ongoing developments in 3GPP TSG SA3 (Service Aspects – Security Group) and 3GPP TSG RAN2 (Radio Architecture Network - Layer 2/3 Group).

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.

The exact throughput requirements will depend on the RLC PDU / MAC SDU size and the number of RLC PDUs / MAC SDUs which may be sent in a single physical layer 10ms frame. In addition, within each 10ms frame, the algorithm will need to be reinitialised (or reinstantiated) for each different bearer and for each transmission direction.

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 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.

5.2.7.2 COUNT

COUNT: a time dependent input.

COUNT[0], COUNT[1], …, COUNT[31]

The length of the COUNT parameter is 32 bits. It is assumed that sychronisation of the keystream will be 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 parameter cannot be specified at present. However, it is assumed to be a 32 bit counter.

5.2.7.3 BEARER

BEARER: the identity of the bearer to be encrypted.

BEARER[0], BEARER[1], …, BEARER[3]

The length o BEARER is 4 bits. The same cipher key may be used for different 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 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.

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[X18-1]

The length of LENGTH is X18 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 format of LENGTH cannot be specified at present since the number and sizes of RLC PDUs / MAC SDUs in each 10ms physical layer frame have not yet been fully specified. However, a maximum RLC PDU / MAC SDU size in the region of 1000 bits has been informally indicated by 3GPP TSG RAN2. 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 in a single 10ms physical layer frame for a given bearer and transmission direction. It may consist of user traffic or signalling data. The structure of the plaintext block cannot be specified at present.

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.

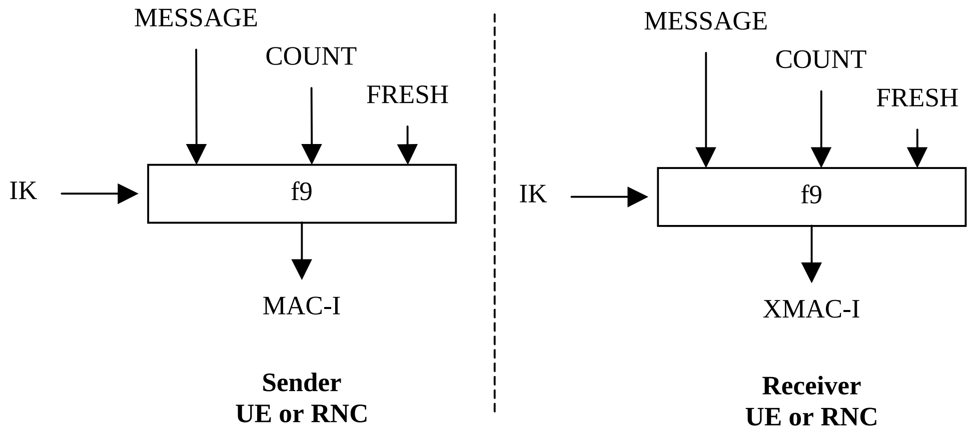


Figure 3: Derivation of MAC-I (or XMAC-I) on a signalling message

The input parameters to the algorithm are the Integrity Key (IK), a time dependent input (COUNT), a random value generated by the network side (FRESH) 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.

The exact position of MAC algorithm in the radio network architecture has not yet been fully specified. The current working assumption is that it will be closely integrated with the ciphering algorithm.

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. 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.

5.3.7.2 COUNT

COUNT: a frame dependent input.

COUNT[0], COUNT[1], …, COUNT[31]

The keystream should be initialised with a time dependent input parameter.

It is assumed that sychronisation of the keystream will be 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 parameter cannot be specified at present. However, it is assumed to be a 32 bit counter.

5.3.7.3 FRESH

FRESH: a random number generated by the RNC.

FRESH[0], FRESH[1], …, FRESH[31]

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 MAC-I (and equivalently XMAC-I)

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

MAC-I[0], MAC-I[1], …, MAC-I[X20-1]

The length of MAC-I is 24 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): User identity confidentiality

A.1 Overview

Figure A**Error! Reference source not found.** illustrates the use of the encryption function f6 to encrypt the IMUI and the sequence for user identity confidentiality (SEQ_UIC) into an EMUI and the use of the decryption function f7 to decrypt the EMUI and retrieve the SEQ_UIC and the IMUI.

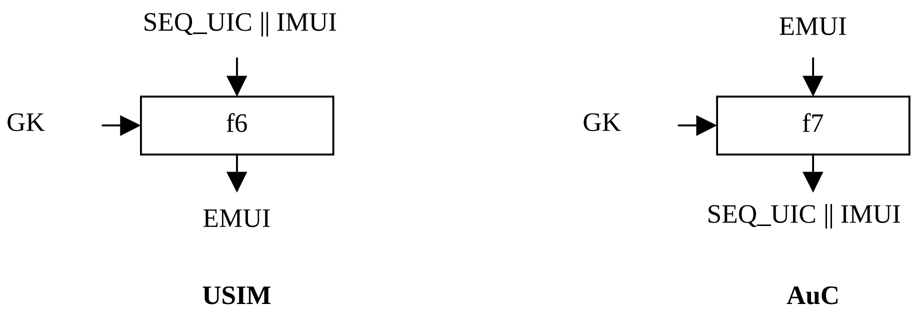


Figure A: Encryption and decryption of the permanent user identity

The mechanism for user identity confidentiality that is described in annex B of [1] requires the following cryptographic functions:

f6	the user identity encryption function;
f7	the user identity decryption function.

A.2 Use

The functions f6 and f7 shall only be used to protect the confidentiality of the user identity when transmitted from USIM to AuC.

A.3 Allocation

The function f6 is allocated to the USIM. The function f7 is allocated to the Authentication Centre.

A.4 Extent of standardisation

The functions f6 and f7 are proprietary to the home environment.

A.5 Implementation and operational considerations

The function f6 shall be designed so that it can be implemented on an IC card equipped with a X1-bit microprocessor running at X2 MHz and with X3 kbits of memory and produce EMUI in less than X11 ms.

The functions f7 shall be designed so that they can be implemented in software in the AuC on a X6-bit microprocessor running at X7 MHz and X8 kbits of memory and produce SEQ_UIC || IMUI in less than X12 ms.

A.6 Type of algorithm

A.6.1 f6

f6: the user identity encryption function

f6: (GK; SEQ_UIC || IMUI) → EMUI

f6 should be a block cipher.

A.6.2 f7

f7: the user identity decryption function

f7: (GK; EMUI) → SEQ_UIC || IMUI

f7 should be a block cipher and the inverse function of f6, in the sense that

x = f7(y; f6(y; x)), for all valid x = SEQ_UIC || IMUI and all valid y = GK.

A.7 Interface

A.7.1 GK

GK: the user group key

GK[0], GK[1], …, GK[X13-1]

The maximum length of the group key GK is X13 bits. The user group key GK is a long term secret key stored in several USIMs and in the AuC.

A.7.2 SEQ_UIC

SEQ_UIC: the sequence for user identity confidentiality

SEQ_UIC[0], SEQ_UIC[1], …, SEQ_UIC[X14-1]

The length of SEQ_UIC is X14 bits. The SEQ_UIC is generated by the USIM and should be different each time so as to prevent traceability of a user.

A.7.3 IMUI

IMUI: the international mobile user identity

IMUI[0], IMUI[1], …, IMUI[X15-1]

The length of the IMUI is X15bits. The IMUI is the permanent identity of the user, stored in the USIM and in the AuC.

A.7.4 EMUI

EMUI: the encrypted mobile user identity

EMUI[0], EMUI[1], …, EMUI[X16-1]

The length of the EMUI is X16 bits.

Annex B: Functions for alternative AKA protocol

B.1 Scope

The mechanism described here achieves mutual authentication and key agreement between the USER (e.g.,USIM) and the AuC in the user's HE, showing knowledge of a secret key K which is shared between and available only to these two parties. The temporary authentication key generated during execution of the protocol is shared with the visited SN/VLR, and can be used subsequently with the local authentication and session key agreement protocol described in section D.2 of Annex D of [1] or with the other local authentication mechanisms described in Annex D [1]. Additionally, the session keys for the first session are created during execution of the protocol.

Architectural detail can be obtained from Annex D of [1].

B.2 Abbreviations

The following abbreviations are complementing the abbreviations defined above in section 3.3 of this document:

KT	Temporary Authentication Key	
RES1	Response to the USER-generated Challenge	(HE/AuC → SN/VLR → USER)
RES2	Response to the Network-generated Challenge	(USER→ SN/VLR → HE/AuC)
RSn[1]	Network Generated Challenge	(HE/AuC → SN/VLR → USER)
Rsu	USER-generated Challenge	(USER → SN/VLR → HE/AuC)
SN	Serving Node	
XRES1	Expected Response to the Network-generated Challenge (RES1)	
XRES2	Expected Response to the User-generated Challenge (RES2)	

B.3 General algorithm requirements

The requirements listed in the body of the document should also apply to the **AKA** protocol. Only the differences will be described in this Annex.

B.4 Functional algorithm requirements

B.4.1 Overview

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

f1 Temporary Authentication Key Generation Function (KT)

f2 Function to Calculate Response to User Challenge (RES1) or the Expected Response to Network

[1] RSn and/or RSu can be a random number or a counter

Challenge (XRES1)

f3 Function to Calculate the Response to the Network Challenge (RES2) or the Expected Response to User Challenge (XRES2)

f4 Function to Derive the Cipher Key (Ck)

f5 Function to Derive the Integrity Key (Ik)

Figure B.1 below illustrates the use of the cryptographic functions f1 – f5 at the HE/AuC, SN/VLR and USER.

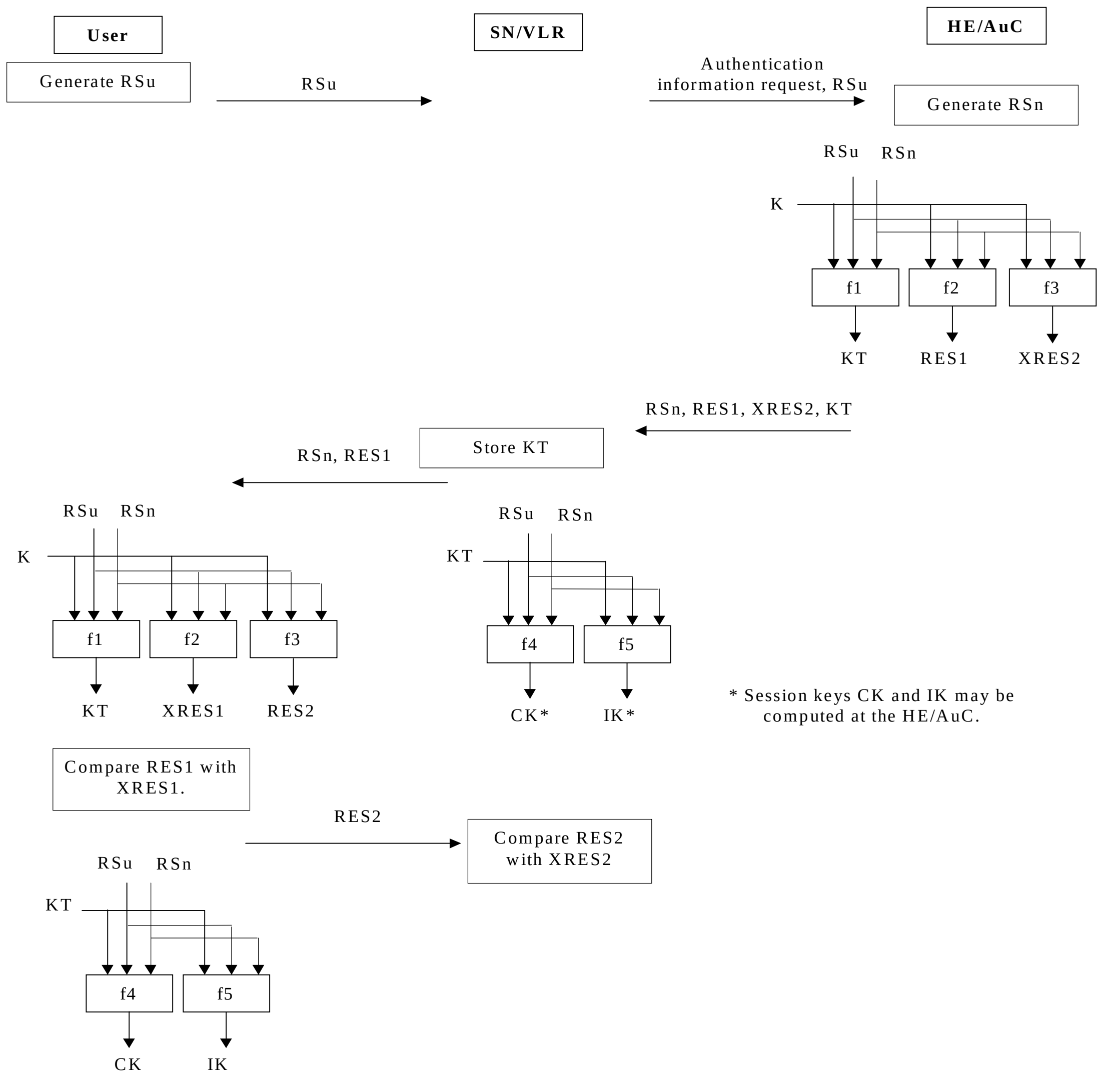


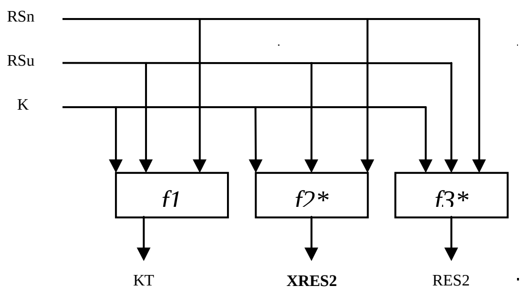
Figure B.1 Functions for generating the Temporary Key, the Ciphering Key, the Integrity Key and the responses to the authentication challenges (X)RES1 and (X)RES2.

The input parameters to the **f1**, **f2**, and **f3**(see <u>note</u>) algorithms are:

- the **K** (Authentication Key)

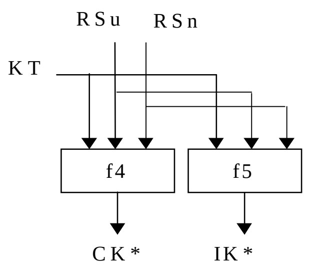
  (<u>note</u>: when authentication is performed <u>locally</u>, between the SN/VLR and the User, the* **K** input for **f2,f3** *is replaced by the* **f1** *derived Temporary Key –* **KT** for deriving the **(X)RES2***)*

- the **RSu** (Random Number – User)
- the **RSn** (Random Number – Network)



The input parameters for the **f4**, **f5** algorithms are:

- the Temporary Key (**KT**) generated by **f1**
- the **RSu**
- the **RSn**



B.4.2 Use

The functions **f1** – **f5** shall be used only to provide mutual entity authentication between the User and the AuC (or the SN/VLR when the temporary authentication key **KT** is shared). The derived keys shall protect the user and the transmission over the radio access link (voice or data). The confidentiality of the user identity should be protected.

B.4.3 Extend of standardisation

The **f2** – **f5** should be standardise by ETSI SAGE. The **f1** is proprietary to the home environment.

B.4.4 Allocation

The functions **f1** – **f5** are allocated to the Home Entity / Authentication Centre (AuC) and the USER (e.g., UIM). The **f2 - f5** can also be allocated to the SN/VLR where the permanent key (**K**) is replaced, for the currently active registration period, by the Temporary Key (**KT**) generated at the HE/AuC by **f1**.

B.4.5 Implementation and operational considerations

ATK implementation shall conform to the same requirements as described in section 5.1.5 of the document.

B.4.6 Type of algorithms

B.4.6.1 f0

This function is a pseudo random number generator, as described above in this document. It is used to generate a network random number (RSn) as well as the USER random number (RSu).

f0: (internal state) → RSn

f0: (internal state) → RSu

B.4.6.2 f1

Authentication function that generates the temporary authentication key used, by the serving system, to generate the session ciphering, the integrity keys or perform local authentication. This function is not shared with the SN/VLR and does not have to be standardized.

f1: (K, RSu, RSn) → KT

It should be computationally unfeasible to derive the K from the knowledge of RSu, RSn. or/and KT.

B.4.6.3 f2

Network authentication function deriving the response and the expected response to an authentication challenge from the AuC.

f2: (K, RSn, RSu) → RES1 (in the AuC)

f2: (K, RSn, RSu) → XRES1 (in the USER terminal)

When a local authentication session key agreement is executed, the **KT** is used in the computation instead of the **K** (the SN/VLR is sending the authentication challenge to the USER).

f2: (KT, RSn, RSu) → RES1 (in the SN/VLR)

f2: (KT, RSn, RSu) → XRES1 (in the USER terminal)

It should be computationally unfeasible to derive the K or KT from the knowledge of RSu, RSn. or/and (X)RES1.

B.4.6.4 f3

Network authentication function deriving the response and the expected response to an authentication challenge from the USER terminal.

f3: (K, RSn, RSu) → RES2 (in the USER terminal)

f3: (K, RSn, RSu) → XRES2 (in the AuC)

When a local authentication session key agreement is executed, the **KT** is used in the computation instead of the **K**.

f3: (KT, RSn, RSu) → RES2 (in the USER terminal)

f3: (KT, RSn, RSu) → XRES2 (in the SN/VLR)

It should be computationally unfeasible to derive the K or KT from the knowledge of RSu, RSn. or/and (X)RES2.

B.4.6.5 f4

Function used to generate a session's ciphering key used to encode the over the air conversation between the user's terminal and the serving system.

f4: (KT, RSn, RSu) → Ck (in the USER terminal)

f4: (KT, RSn, RSu) → Ck (in the SN/VLR)

It should be computationally unfeasible to derive the K or KT from the knowledge of RSu, RSn. or/and Ck.

B.4.6.6 f5

Function used to generate a session's integrity key used to encode the over the air signaling between the user's terminal and the serving system.

f5: (KT, RSn, RSu) → Ik (in the USER terminal)

f5: (KT, RSn, RSu) → Ik (in the SN/VLR)

It should be computationally unfeasible to derive the K or KT from the knowledge of RSu, RSn. or/and Ik.

B.4.7 Interface

The following additions are proposed to the interfaces described above in this document.:

B.4.7.1 KT

KT: the user temporary authentication key.

The length of the KT should be the same as specified for K above.

B.4.7.2 RSn and RSu

RSn, RSu: random numbers

The length of RSn and RSu should be the same as specified for the RAND above.

B.4.7.3 RES1, XRES1, RES2, XRES2

RES1 or XRES1: authentication challenge from the network or the expected response from the user.

RES2 or XRES2: authentication challenge from the USER or the expected network response.

The length of those parameters should be the same as specified for the RES above.

Annex C: History

Document history	
4 February 1999	Scope (draft) at 3GPP TSG-SA WG3 #1
24 February 1999	Scope and Contents (draft) to 3GPP TSG-SA WG3 mailing list
15 March 1999	Version 0.0.1 to 3GPP TSG-SA WG3 #2
26 March 1999	Version 0.0.2 at 3GPP TSG-SA WG3 #2
28 April 1999	Version 0.0.3 to ESTI SAGE (and 3GPP TSG-SA WG3 mailing list)
7 May 1999	Version 0.0.4 to 3GPP TSG-SA WG3 #3
19 May 1999	Version 0.0.5 to 3GPP TSG-SA WG3 mailing list (for editing meeting)
27 May 1999	Version 0.0.9 to 3GPP TSG-SA WG3 Adhoc meeting attendees
31 May 1999	Version 0.1.0 to 3GPP TSG-SA WG3 mailing list
14 June 1999	Version 0.1.1 to 3GPP TSG-SA WG3 #4
18 June 1999	Version 0.2.0 at 3GPP TSG-SA WG3 #4

Annex D: Unspecified values

Reference	**Meaning**	**Range**	**Source**
X1	Bus width of the USIM processor (bit)		TSG T WG3
X2	Clock speed of the USIM processor (MHz)		TSG T WG3
X3	Memory size of the USIM (kbits)		TSG T WG3
X4	Response time for AK, MAC-A and RES (ms)		TSG SA WG2
X5	Response time for CK and IK (ms)		TSG SA WG2
X6	Bus width of the AuC processor (bit)		TSG CN
X7	Clock speed of the AuC processor (MHz)		TSG CN
X8	Memory size of the AuC (kbits)		TSG CN
X9	Response time for authentication vector in AuC (ms)		TSG SA WG2
X10	Length of sequence number (bits)	32—64	TSG SA WG3
X11	Response time for EMUI computation in the USIM (ms)		TSG SA WG2
X12	Response time for SEQ_UIC || IMUI in the AuC (ms)		TSG SA WG2
X13	Length of the group key (bits)	128	TSG SA WG3
X14	Length of SEQ_UIC (bits)	32	TSG SA WG3
X15	Length of IMUI (bits)		TSG SA
X16	Length of EMUI (bits)	128	TSG SA WG3
X17	Number of gates required for hardware implementation of ciphering algorithm	10 000	TSG T WG3 TSG CN
X18	Length of the field LENGTH for ciphering (bits)		TSG RAN WG2
X19	Maximum length of a signalling message (bits)		TSG SA WG3 TSG RAN WG2
X20	Length of MAC-I (bits)	24	TSG SA WG3
X21	Length of RES and XRES (bits)	32-128	TSG SA WG3