Technical Specification Group Services and System Aspects
Meeting #7, Madrid, Spain, 15-17 March 2000

TSGS#7(00)0049

Source:	**SA WG3**
Title:	**General Report from ETSI SAGE on the Design, Specification and Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms**
Document for:	**Information**
Agenda Item:	**5.3.2**

The attached report was received by ETSI SAGE and is forwarded to TSG SA as supporting information to SP-000005 from SA WG3 (LS to ETSI SAGE (cc SA) on Delivery of algorithm specifications).

SA WG3 S3-000105
ETSI SAGE 3GPP Task Force (99) 38

ETSI SAGE 3GPP Standard Algorithms Task Force

Public Report

Security Algorithms Group of Experts (SAGE); General Report on the Design, Specification and Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms

---

DRAFT VERSION – 1.0

Date: 1999-12-22

Reference

Keywords
3GPP, security, SAGE, algorithm

ETSI Secretariat

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - 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

X.400
c= fr; a=atlas; p=etsi; s=secretariat

Internet
secretariat@etsi.fr
http://www.etsi.fr

Contents

Intellectual Property Rights 4
Foreword 4
1 Scope 5
2 References 5
3 Abbreviations 6
4 Structure of this report 7
5 Background to the 3GPP confidentiality and integrity algorithms 7
6 SAGE TF 3GPP work plan 8
7 Outline of algorithm requirements specification 9
7.1 f8 – Confidentiality algorithm 9
7.2 f9 – Integrity algorithm 13
7.3 Generic requirements for 3GPP cryptographic functions and algorithms 15
8 Algorithms design 16
8.1 Design criteria 16
8.1.1 Algorithm Basics 16
8.1.2 Performance and Implementation Requirements 16
8.1.3 Starting Point for the Designs 17
8.1.4 Particular Cryptographic Criteria 17
8.2 Design methodology 18
8.3 Specification and test data 19
9 Algorithm evaluation 20
9.1 Evaluation criteria 20
9.1.1 Mathematical Evaluation Criteria 20
9.1.1.1 Analysis of various components of KASUMI 20
9.1.1.2 Analysis of KASUMI as a generic 64-bits blockcipher 20
9.1.1.3 Analysis of the encryption and integrity modes 21
9.1.2 Statistical Evaluation Criteria 21
9.1.1.2 Statistical Tests on the UMTS Confidentiality and Integrity Algorithm 21
9.1.2.2 Statistical Tests on the Blockcipher KASUMI 22
9.1.2.3 Building Blocks of the Blockcipher KASUMI 24
9.2 Method of evaluation 24
9.3 Public Evaluation report 25
9.4 Conclusion of evaluation 25
10 Release of algorithm, specification and test data by SAGE 26
11 Export control aspects 26

Intellectual Property Rights

ETSI has not been informed of the existence of any Intellectual Property Right (IPR) which could be, or could become essential to the present document. However, pursuant to the ETSI Interim IPR Policy, no investigation, including IPR searches, has been carried out. No guarantee can be given as to the existence of any IPRs which are, or may be, or may become, essential to the present document.

Foreword

This Report has been produced by ETSI SAGE Task Force for the design of the Standard 3GPP Confidentiality and Integrity Algorithms (SAGE TF 3GPP).

The work described in this report was undertaken in response to a request made by 3GPP.

1 Scope

This report is a description of the work undertaken by SAGE Task Force for the design of the standard 3GPP Confidentiality and Integrity Algorithms (SAGE TF 3GPP), and to formally approve the release of these algorithms to 3GPP.

With regard to the design of the algorithms, the scope of this report is confined to a description of the design criteria, the design methodology and an outline of the content and structure of the specification and test data documents.

The standard 3GPP Confidentiality and Integrity Algorithms are based on a Block Cipher named **KASUMI**. The algorithms specification and associated test data are documented in the Specification of the 3GPP Confidentiality and Integrity Algorithms which consists of the following four documents.

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 1: f8 and f9 Algorithm Specifications

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 2: **KASUMI** Algorithm Specifications

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 3: Implementors' Test Data

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 4: Design Conformance Test Data

With regard to the evaluation of the algorithm, the scope of this report is restricted to a description of the evaluation criteria, the method of evaluation and the main conclusions from the evaluation that led to the Task Force approving the specification. A detailed summary of conclusions of the evaluation is provided in a public evaluation report [3] produced by the Task Force.

2 References

For the purposes of this report, the following reference apply:

[1] 3rd Generation Partnership Project: technical Specification Group Services and System Aspects; 3G Security; Cryptographic Algorithm Requirements (3G TS 33.105 version 3.1.0)

[2] 3rd Generation Partnership Project: technical Specification Group Services and System Aspects; 3G Security; Security Architecture (3G TS 33.102 version 3.2.0)

[3] ETSI SAGE 3GPP Standards Algorithms Task Force: Report on the Evaluation of 3GPP Standard Confidentiality and Integrity Algorithms

3 Abbreviations

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

3GPP	3rd Generation Partnership Project
f8	UMTS confidentiality (encryption) algorithm
f9	UMTS integrity algorithm
MISTY	Japanese block cipher algorithm which was the basis for the design of KASUMI
KASUMI	Block cipher algorithm which was the basis for f8 and f9
SAGE	Security Algorithms Group of Experts
SAGE TF 3GPP	SAGE Task Force for the design of the standard 3GPP Confidentiality and Integrity Algorithms
UMTS	Universal Mobile Telecommunications System

4 Structure of this report

This material presented in this report is organised in the subsequent clauses, as follows:

- clause 5 provides background information on standard 3GPP Confidentiality and Integrity Algorithms and KASUMI;

- clause 6 provides an outline of the work plan adopted by SAGE TF 3GPP to design and evaluate the algorithms and to approve the algorithms specification and associated test data for release 3GPP;

- clause 7 consists of a summary of the main points in the algorithm requirements specification produced by 3GPP TSG SA3;

- clause 8 describes the way in which SAGE TF 3GPP designed the algorithm and produced the specification and associated test data;

- clause 9 gives an overview of the evaluation work carried out by SAGE TF 3GPP and other parties and the conclusions of the evaluations;

- clause 10 summarises the result of the SAGE TF 3GPP internal approval procedures;

- clause 11 outlines export control issues especially for the confidentiality algorithm.

5 Background to the 3GPP confidentiality and integrity algorithms

Within the mobile communication system UMTS specified by 3GPP there is a need to provide security features. These security features are realised with the use of cryptographic functions and algorithms. In total 3GPP identified the need for 9 cryptographic algorithms and functions (ref. [2]). It also was decided that two cryptographic algorithms, f8 (the confidentiality algorithm) and f9 (the integrity algorithm) need to be standardised. The requirement specifications for the cryptographic algorithms were drafted by 3GPP (ref. [1]).

Then ETSI SAGE was asked to design the algorithms. To carry out this work ETSI SAGE set up a Task Force (SAGE TF 3GPP) which with the assistance of a number of other parties designed and specified the algorithms.

Because of the short time scales it was decided to base the algorithms on an existing algorithm which had already undergone some evaluation. ETSI SAGE and 3GPP TSG SA3 agreed to select the algorithm MISTY (ref. http://www.mitsubishi.com/ghp_japan/misty/index.htm) as a starting point.

6 SAGE TF 3GPP work plan

After some preparatory work by ETSI SAGE the SAGE TF 3GPP formally started work mid August 1999. The SAGE TF 3GPP consisted of the regular SAGE members, the designer of the MISTY algorithm and three manufacturers from 3GPP. The work was funded by 3GPP and the and three manufacturers.

The design of the algorithms and a complete set of specification documents were finalised mid November 1999. It was decided by 3GPP that the algorithms should be evaluated, during a one month period by three groups of independent evaluators, all consisting of well known cryptologists.

The three groups of independent evaluators were:

- A consortium led by Leuven University, Leuven, Belgium
- Cryptolog, Paris, France
- Royal Holloway College, University of London, UK

The results of these evaluations were reviewed by SAGE TF 3GPP before the final algorithms specifications were released to 3GPP.

The total resource budget for the SAGE TF 3GPP work funded by 3GPP was 550 man-days.

Of this budget, approximately 220 days were allocated to the design and specification of the algorithm and 230 days to the evaluation. The rest was spent on specification testing, liaison and management procedures. In addition to the 230 days spent on evaluation three manufacturers (Ericsson, Motorola and Nokia) spent a significant amount of time on the evaluation, on their own budget. The estimated time spend is 35 days by Ericsson, 18 days by Motorola and 30 days by Nokia.

The SAGE TF 3GPP work was thus carried out by eleven organisations, which were divided into two teams: a design team and an evaluation team. The allocation of budget funded by 3GPP over the participating organisations was spread such that four organisations each had about 15-17 % of the budget, three each had 10–13% and one had 4%.

The work was divided into two main tasks:

- Design and Specification testing (approximately 41% of the budget funded by 3GPP);
- Evaluation (approximately 44% of the budget funded by 3GPP);

and three smaller tasks

- Management, export control issues, liaison (approximately 15% of the budget funded by 3GPP).

7 Outline of algorithm requirements specification

The requirements for the f8 and f9 algorithms were specified by 3GPP TSG SA3 in: 3rd Generation Partnership Project: technical Specification Group Services and System Aspects; 3G Security; Cryptographic Algorithm Requirements (3G TS 33.105 version 3.1.0)

The functional requirements for the algorithm as formulated by ETSI SMG10 are summarised in the following sections.

7.1 f8 – Confidentiality algorithm

The requirements for this algorithm are given in section 5.2 of [1] and are summarized below (in italics).

5.2.1 Overview[1]

The mechanism for data confidentiality of user data and signalling data that is described in {relevant 3GPP standard document} 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.

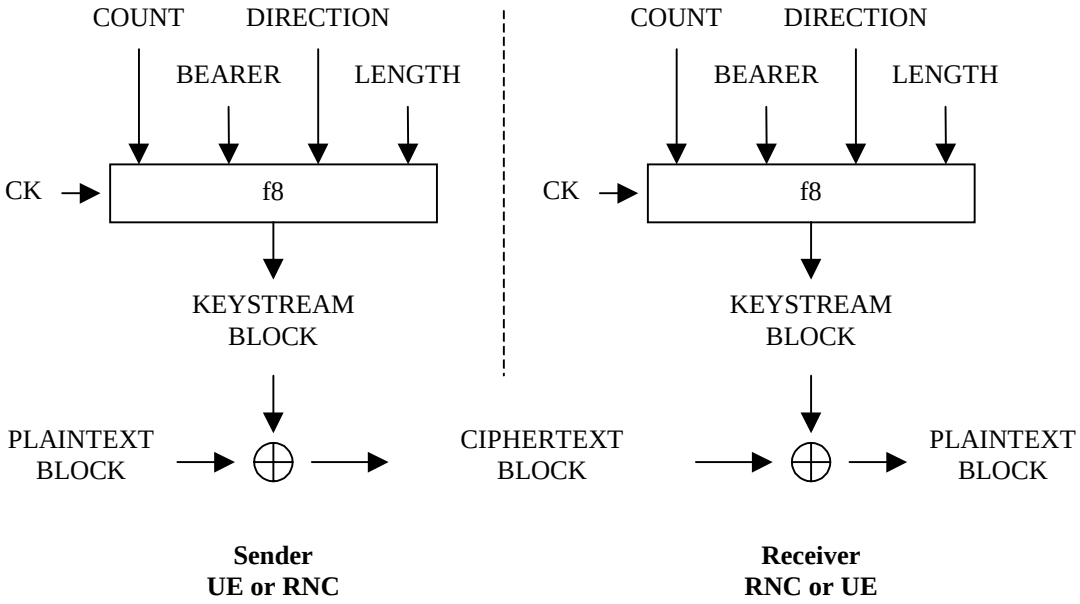


Figure 1: Ciphering user and signalling data transmitted over the radio access link

The input parameters to the algorithm are the Cipher Key (CK), a time dependent input (COUNT), 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

[1] Note that the section numbers 5.2.* refer to the original text in [1]

(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.

1. *RLC-transparent mode:*

- *New keystream block required every physical layer frame (10ms)*
- *Maximum number of bits per physical layer frame of 5114 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 every RLC frame (minimum 156μs)*
- *Maximum number of bits per UM RLC frame of 1016 bits (ongoing specification work in TSG-R2 could extend this to 5000 bits)*
- *Minimum number of bits per UM RLC frame of 16 bit.*
- *Granularity of 8 bit on all possible intermediate values*

3. *For AM RLC mode:*

- *New keystream block required every RLC frame (minimum 156μs)*

- *Maximum number of bits per AM RLC frame of 1024 bits (ongoing specification work in TSG-R2 could extend this to 5000 bits)*
- *Minimum number of bits per AM RLC frame of 24 bit.*
- *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 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[2]*. 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.

---

2 In a later stage the length of BEARER was changed to 5 bits and denoted as *BEARER[0], BEARER[1], …, BEARER[4]*

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.

7.2 f9 – Integrity algorithm

The requirements for this algorithm are given in section 5.3 of [1] and are summarized below (in italics).

5.3.1 Overview[3]

The mechanism for data integrity of signalling data that is described in {relevant 3GPP Standard} requires the following cryptographic function:

f9 *UMTS integrity algorithm.*

Figure 2[4] *illustrates the use of the function f9 to derive a MAC-I from a signalling message.*

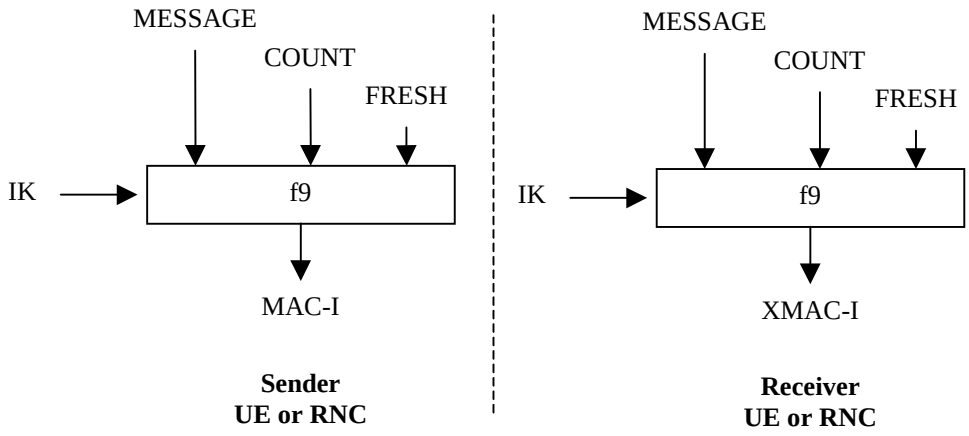


Figure 2: Derivation of MAC-I (or XMAC-I) on a signalling message

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.

---

[3] Note that the section numbers 5.3.* refer to the original text in [1]

[4] Note that the input “DIRECTION” is accidentally missing from this figure

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 standardised.

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-I

COUNT-I: a frame dependent input.

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

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

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. The length of COUNT-I parameter is assumed to be 32 bits.

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 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.

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.

7.3 Generic requirements for 3GPP cryptographic functions and algorithms

In section 4 of [1] generic requirements are given for all 3GPP cryptographic functions and algorithms. These are summarized below (in Italics).

4.1 Resilience[5]

The functions should be designed with a view to its continued use for a period of at least 20 years. Successful attacks with a workload 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.

4.2World-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] Note that the section numbers 4.* refer to the original text in [1]

8 Algorithms design

Based on the requirements and fixed starting points SAGE TF 3GPP established the following essential design criteria.

8.1 Design criteria

8.1.1 Algorithm Basics

Confidentiality Algorithm f8

- f8 is a synchronous binary stream cipher.
- Inputs to the keystream generator algorithm are the cipher key CK (128 bits), a time-dependent input COUNT (32 bits), a 4-bit BEARER identifier and a 1-bit DIRECTION identifier. (The length of keystream required is also considered as an input to the algorithm, but it only affects the number of keystream bits to be returned, not their values.)

Integrity Algorithm f9

- f9 is a MAC algorithm.
- Inputs to the algorithm are the integrity key IK (128 bits), a frame counter COUNT (32 bits), a "random" number FRESH (32 bits) generated by the RNC, and a variable length message.[6]
- The output from f9 is a 32-bit MAC value.

8.1.2 Performance and Implementation Requirements

Confidentiality Algorithm f8

Reference [1] includes the following stated requirements:

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 re-instantiated) 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).

Further input [Liaison Statement from RAN WG2 to SA3 on Ciphering algorithm Requirements (S3-99228)] indicates that:

---

[6] Later it was agreed with 3GPP SA3 to introduce a Direction bit (1 bit) as input

- the maximum possible length of a data unit to be encrypted is 5000 bits, with a granularity of 1 bit;
- there will be no more than 64 data units per 10ms.

Integrity Algorithm f9

Reference [1] is less explicit about implementation requirements for f9. It states:

- The algorithm should be designed to accommodate a range of implementation options including hardware and software implementations.
- The current working assumption is that it will be closely integrated with the ciphering algorithm

We assume that, if the MAC algorithm processes data roughly as fast as the encryption algorithm, and can be implemented with only modest extra complexity on top of what is needed for the encryption algorithm, then all will be well.

Further input [Liaison Statement from RAN WG2 to SA3 on Ciphering algorithm Requirements (S3-99228)] indicates that, as for the encryption algorithm:

- the maximum possible length of a data unit to be MAC-ed is no more than 5000 bits, with a possible granularity of 1 bit;
- there will be no more than 64 data units per 10 ms.

8.1.3 Starting Point for the Designs

The SAGE TF 3GPP decided to build both algorithms from a single block cipher, which should be closely related to one of the MISTY algorithms. (MISTY has been selected in preference to other block ciphers primarily because of its provable security aspects and its suitability for hardware implementation.)

8.1.4 Particular Cryptographic Criteria

There are general strength criteria that are required of any stream cipher and any MAC function; these will not all be listed here. Below, however, are a few points that should particularly be borne in mind.

MISTY

- The most successful attacks on (simplified, reduced round) MISTY are higher order differential attacks and interpolation attacks. Any modifications to MISTY must not increase the algorithm's susceptibility to these attacks — ideally, they should strengthen it against them.

General

- These algorithms will use short-term keys presented by the SIM to the phone. There is therefore no requirement for resistance to differential power analysis (which is aimed at extracting a long-term key from supposedly secure storage).
- Keys are always randomly generated, so related key attacks are of very little practical significance (but for a published algorithm preferably we should avoid even academic attacks.)

Confidentiality Algorithm f8

- The most popular way of building a stream cipher from a block cipher is to use it in output feedback mode. This approach carries a (very small) risk of short keystream cycles. It would be desirable to remove this risk altogether.
- It is also desirable to avoid any possibility of the keystream generator getting itself into exactly the same state — and hence generating the same keystream from then onwards — at any two points in different frames.

Integrity Algorithm f9

- For absolute security, it must be impossible for an attacker to intercept one {Message, MAC} pair, modify the message in any way, and have the MAC either unchanged or modified in a way he can predict (e.g. linearly). In particular, therefore, the padding applied to a message to bring it to a whole number of blocks must be such that it is not feasible to construct two messages that are identical after padding.

8.2 Design methodology

The algorithms were designed using the iterative, interactive and phased approach that is normally applied for the design of ETSI SAGE algorithms. The design process is summarised below.

- **Phase 1:** The starting points for the algorithms and design criteria were agreed. The design team then produced a first design proposal for a modified MISTY algorithm. This algorithm was called **KASUMI** (the Japanese word for MISTY). This was presented for consideration by the SAGE TF 3GPP evaluation team.

  In addition the original MISTY algorithm was sent to 3GPP participants with a request to review its performance and implementation complexity characteristics. This resulted in positive responses and no indications that the algorithm would be too complex in implementation or slow in operation.

- **Phase 2:** During a meeting the results of the evaluation were discussed. Also the possible use of KASUMI to implement the f8 and f9 algorithms was discussed. Based on these results, the design team revised the KASUMI design to produce a second design proposal for the algorithm. The evaluation team again reviewed the revised KASUMI design and f8 and f9 proposals.

- **Phase 3:** The results of the second evaluation were discussed during a second SAGE TF 3GPP meeting. During this meeting the design for the KASUMI and f8 and f9 algorithms was fixed, except for some small details.

- **Phase 4:** The KASUMI, f8 and f9 algorithms were fully fixed. A final round of statistical tests on the algorithms was carried out. The specification documents were drafted and two parties independently carried out a specification testing to check the correctness and completeness of the specification.

The specification document and statistical test data, as well as a summary of the evaluation undertaken by the SAGE TF 3GPP were then made available to three groups of independent evaluators.

- **Phase 5:** During 4 weeks the algorithms were evaluated by a three groups of independent evaluators. This resulted in three evaluation reports. These reports we reviewed by the SAGE TF 3GPP. After this review the algorithms specifications were finalized.

8.3 Specification and test data

The algorithm specification and associated test data are documented in the Specification of the 3GPP Confidentiality and Integrity Algorithms which consists of the following four documents.

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 1: f8 and f9 Algorithm Specifications

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 2: **KASUMI** Algorithm Specifications

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 3: Implementors' Test Data

- Specification of the 3GPP Confidentiality and Integrity Algorithms

  Document 4: Design Conformance Test Data

Document 1 is normative and contains the formal specification of the functional elements of f8 and f9 algorithms. There are two informative annexes to Document 1. The first annex consists of illustrative diagrams to aid understanding of the specification. The second annex consists of an example program listing of the algorithm in 'C'.

Document 2 is normative and contains the formal specification of the functional elements of **KASUMI**, the algorithm on which both f8 and f9 are based. There are two informative annexes to Document 2. The first annex consists of illustrative diagrams to aid understanding of the specification. The second annex consists of an example program listing of the algorithm in 'C'.

Document 3 is informative and provides design conformance test data designed to help verify implementations of the **KASUMI**, f8 and f9 algorithms. The document identifies the relevant intermediate points in the algorithms where test data is provided. Then it gives input, internal and output parameters at these points, and provides different sets of test data listings.

Document 4 is informative and provides test data designed to help verify the correct functioning of the f8 and f9 algorithms seen as a 'black box'. The document identifies the input and output interfaces and provides a number of test sets for the different modes of operation of the algorithms. The test sets are designed in such a way that all elements of any functions in the algorithms are used at least once.

9 Algorithm evaluation

9.1 Evaluation criteria

The Algorithms Requirements as summarised in section 7 and Design Criteria as listed in section 8 led to two sets of evaluation criteria: one set for the mathematical evaluation and one set for the statistical evaluation.

9.1.1 Mathematical Evaluation Criteria

The Mathematical Evaluation Criteria are detailed below.

9.1.1.1 Analysis of various components of KASUMI

This part of the analysis will focus on algorithm components such as:

- the S7 and S9 S-Boxes
- the FL function
- the FI function
- the FO function
- the key generation and key scheduling

The algebraic, statistical, or pseudo randomness properties of these components which seem most directly related to the security of the KASUMI cipher will be investigated.

9.1.1.2 Analysis of KASUMI as a generic 64-bits blockcipher

This will represent the main part of the mathematical analysis. The resistance of KASUMI and simplified versions of KASUMI (i.e. KASUMI with a reduced number of rounds and KASUMI without any FL function) against various categories of attacks will be investigated.

One can (informally) describe as an attack of a 64-bits blockcipher any method enabling an adversary provided with less than 2^{64} adaptively chosen plaintexts or ciphertexts under an unknown key to predict any additional plaintext or ciphertext pair with a non negligible advantage over the situation where the blockcipher would have been replaced by a truly random permutation.

Types of attack to be considered include:

- **Meet in the middle attacks**: split the key in two, perform some sort of exhaustive listing of the effects of each half, and then look for a match.

- **Differential attacks**: finding pairs of input with a certain relationship (e.g. constant XOR) that (depending on the key) probabilistically yield output pairs with a certain relationship (e.g. constant XOR), and hence deducing some information about the key. Since KASUMI offers some provable resistance against pure differential cryptanalysis,

the analysis will focus on the investigation of variants of differential attacks such as miss in the middle attacks [Biham-Shamir-Byrukov], boomerang attacks [Wagner], truncated differentials [Knudsen], etc.

- **Weak keys**: membership of a reasonably large class of keys detectable because of some special or incomplete functionality they cause within the algorithm.

- **(Probabilistic) linear factors**: complementing a set of key bits (probabilistically) adds a constant to the sum of a set of output bits, hence reduce size of key that needs to be searched by one bit.

- **Linear cryptanalysis**: find high-probability parity of the sum of some input, output and key bits, and hence deduce one bit of information about the key. Since KASUMI offers some provable resistance against pure linear cryptanalysis, the analysis will focus on the investigation of variants of linear attacks such as linear–differential cryptanalysis [Langford-Hellman], higher order cryptanalysis [Lai], and other statistical cryptanalysis methods [Vaudenay, Murphy, Gilbert...].

- **Interpolation attacks**: exploiting the low degree of the algebraic relation between some input (resp. output) and intermediate data to infer some keybits relating the output (resp. input) and the intermediate data.

- **Partial key guess**: guessing a small part of the key makes one of the above attacks feasible.

9.1.1.3 Analysis of the encryption and integrity modes

This part of the analysis consists in investigating the strength of constructions used for deriving the f8 and f9 algorithms from the KASUMI blockcipher – in order to make sure that the f8 and f9 construction do not substantially deviate from the following ideal requirements:

- (f8) : There should be no efficient test enabling an adversary to distinguish the f8 algorithm (as seen as a pseudorandom function generator associating a key with a mapping from the IV set to the output sequences set) from a truly random function generator.

- (f9) : The integrity algorithm should resist existential forgery by adaptive adversary, i.e. it should be computationally infeasible for an adversary to infer any additional MAC value of an N+1th message from a set of N adaptively obtained MAC values corresponding to N messages.

The operational context of use of the f8 and f9 algorithms (repetition of IV values, redundancy of the plaintext, etc.) will be as much as possible taken into account in the analysis.

9.1.2 Statistical Evaluation Criteria

The Statistical Evaluation Criteria fall in to two categories: those on the f8 and f9 mode and those on KASUMI.

9.1.1.2 Statistical Tests on the UMTS Confidentiality and Integrity Algorithm

9.1.1.2.1 Streamcipher Mode (f8)

Algorithm f8 is a keystream generator. The produced keystream is used to encrypt the plaintext by XORing plaintext and keystream bit by bit. Thus it is obvious that streamcipher tests on a lengthy sequence produced by f8 will be performed.

Scenario 1: Streamcipher Tests on a Long Keystream Sequence

For these tests one chooses CK, COUNT, BEARER and DIRECTION (defined in [1]) randomly but fixed and increases an internal value BLKCTR [3] by one for each blockcipher encryption, starting with zero. The iterated blockcipher encryptions are performed say 32,768 times, such that one gets a sequence of 2,097,152 bits. This covers 15 out of 27 bits of BLKCTR[7]. On this keystream sequence the following streamcipher tests are performed:

- Frequency test
- Overlapping *m*-tuple test
- Gap test
- Run test
- Coupon-Collector's test
- Universal Maurer test
- Poker test
- Correlation test
- Rank test
- Linear-complexity test
- Ziv-Lempel complexity test
- Collision test
- Run test II
- Maximum-order-complexity test

The results of these tests are evaluated by appropriate statistics, e.g. chi-square-statistic. The algorithm passes a test if there is no significant deviation between the examined sequence and a random sequence.

Scenario 2: Streamcipher Tests on a Concatenation of Small Keystream Sequences

In reality algorithm f8 won't be used to produce a very long keystream sequence but many small keystream sequences which are used to encrypt a data packet or physical layer frame. For the next packet (or frame respectively) a new sequence will be produced by f8. Thus the data stream is encrypted by a concatenation of small separately produced keystream sequences which are at most 1024 (or 1016) bits long due to [4]. Tests were carried out to take this in to account.

9.1.1.2.2 CBC-MAC Mode (f9)

For the integrity of signalling data it is essential that the MAC depends on every bit of the input. To prove this by a statistical (or heuristical) test one could use the Avalanche test (Dependence test) to show that when one input bit changes about half of the MAC bits also change. This test generates two matrices: the dependence matrix describes in row *i* and column *j* how often the output bit *j* changes when the input bit *i* is toggled. The distance matrix describes in row *i* and column *j* how often complementing the i^{th} input bit results in a change of *j* output bits.

Tests were carried out according to three scenarios. In the first scenario one has a fixed message and a fixed initialisation vector (IV) and toggles the bits of IK. In the second scenario IK and the message is fixed an one toggles the bits of the IV. In the third scenario all the input parameters are fixed an one bit of the message changes from time to time.

9.1.2.2 Statistical Tests on the Blockcipher KASUMI

[7] In reality BLKCTR will only be 7 bits long, as messages are limited to 5000 bits

9.1.2.2.1 Dependence Test

For the algorithm itself the Avalanche criteria will be tested. When one bit of the 64 bit input block is toggled about 32 bits of the output block shall change. When one bit of the 128 bit key is toggled about 32 bits of the output block shall change provided the same input block is used. Section 0 contains a detailed description of the Avalanche test.

Scenario 1: Toggle bit of the input block

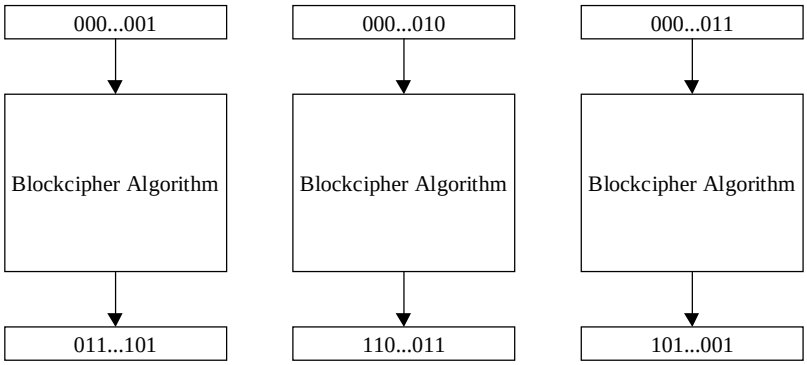
A fixed key and a starting value for the input block are chosen randomly. The dependence test is performed where the bits of the input block are toggled. The resulting matrices are evaluated statistically.

Scenario 2: Toggle bit of the key

A fixed input block and a starting value for the key are chosen randomly. The dependence test is performed where the bits of the key are toggled. The resulting matrices are evaluated statistically.

9.1.2.2.2 Streamcipher Tests on Sequence Generated from Redundant Data

The following test proves statistically that the blockcipher algorithm destroys redundancy in the input data. The key for the algorithm is chosen arbitrarily. A sequence of 64 bit blocks is encrypted by consecutive applications of the blockcipher algorithm in ECB-mode, where between two encryption operations the input block is increased by one, starting with the all-zero block.



The output blocks are concatenated and on the resulting sequence the following tests are applied:

- Frequency test
- Overlapping *m*-tuple test
- Gap test
- Run test
- Coupon-Collector’s test
- Universal Maurer test
- Poker test
- Correlation test
- Rank test
- Linear-complexity test
- Ziv-Lempel complexity test
- Collision test
- Run test II
- Maximum-order-complexity test

9.1.2.2.3 Randomizing Property of the Blockcipher

The following test determines how many rounds of the cipher are necessary to destroy redundancy in an input sequence. The input sequence is once more the sequence of the last section, i.e. starting with the 64 bit all-zero block and consecutively increasing the last block by one. A key is chosen arbitrary and the input sequence is encrypted by the first round of the blockcipher in ECB-mode. The output sequence is evaluated by the Universal Maurer test. This test can be seen as a measure of redundancy since its results are closely related with the per-bit entropy of the sequence. In the sequel the input sequence is encrypted by the first two rounds, the output sequence is evaluated by the Universal Maurer test, and so on, up to the maximum number of rounds which were considered.

9.1.2.3 Building Blocks of the Blockcipher KASUMI

The building parts of the blockcipher have also been evaluated statistically. These parts include the look-up tables, the function FI with 16 bit input/output, the function FO with 32 bit input/output and the function FL with 32 bit input/output.

9.1.2.3.1 S-Box Tests on look-up tables

Each of the look-up tables can be seen as function S: $GF(2)^m \rightarrow GF(2)^m$. The function again can be seen as vectors $S(x) = (s_1(x),\ldots,s_m(x))$ of function s_i: $GF(2)^m \rightarrow GF(2)$ for $1 \leq i \leq m$. On the functions the following tests were carried out:

- Linear Approximation
- Linear Factors Test
- Cycles of the S-Boxes
- The Dependence Test on the S-Boxes
- Differential Tests on the S-Boxes

9.1.2.3.2 The Function FI

The function FI has 16 bit input and 16 bit output. For this function the linear approximation test and the dependence test can be performed.

9.1.2.3.3 The Function FO

The function FO has 32 bit input and 32 bit output. For this function the linear approximation test and the dependence test can be performed.

9.2 Method of evaluation

The evaluation and design teams worked independently during phase 1 and phase 2 (see 8.2) of the work. Only at the start of phase 2 and phase 3 there was interaction to discuss the evaluation results and the design changes required.

During phases 3 and 4 there was a closer co-operation between the design and evaluation teams and the final (minor) modifications were discussed and agreed together.

The methods employed by the evaluation team may be summarized as follows:

- during the first, second and third phase of the work, a detailed mathematical analysis of the algorithm and its component functions as well as statistical analysis of the output of the algorithms and their component functions in relation to the input and the key;

- final round of extensive statistical analysis of the final design in which the statistical properties of the algorithm output were tested in relation to the input and the key;

- four week evaluation of the final design by three independent groups of qualified evaluators, which were selected by 3GPP TSG SA3 from the academic world.

Two parties not directly involved in the design and evaluation teams also evaluated the adequacy of the specification. To this end, these parties made independent simulations of the algorithm from the specification and confirmed these against the test data.

9.3 Public Evaluation report

The public evaluation report [3] is a summary of all results of the complete design and evaluation process. It provides the main conclusions of the evaluation work carried out by the SAGE TF 3GPP as well as the conclusions of the independent evaluations.

9.4 Conclusion of evaluation

A detailed description of the evaluation results can be found in [3]. The conclusion of the evaluation was formulated as follows [3, section 9.8].

> The 3GPP confidentiality and integrity algorithms have been subject to an extensive mathematical and statistical review in order to reveal any weakness in the design. This work has been conducted by the task force itself, by additional manufacturers with competence in the field and by three independent parties. The work has involved some of the leading experts in the field. *The general conclusion is that the algorithms are based on sound design principles, and no practical attacks were found. The algorithms are well fitted for their intended use.*
>
> The algorithms have specifically been designed for use within the 3GPP context. It has not been the intention to increase the security margins in order to develop general-purpose algorithms for multiple unknown applications. The design is a careful trade-off providing full strength algorithms and efficient implementation and use in the next generation mobile systems.
>
> The 3GPP algorithms have been designed to resist a suite of well-known cryptanalytic attacks. However, one can never prove that a cryptographic algorithm will resist new attacks in the future. Due to this fact and the very limited time span that was available for the work, the task force will propose that the results from this report are reviewed on a regular basis. A basic review of the offered security and usability of the 3GPP confidentiality and integrity algorithms should be conducted every five years.

10 Release of algorithm, specification and test data by SAGE

SAGE TF 3GPP approval for release

Prior to release of the specification of the f8 and f9 algorithms and their test data, the following approvals were gained.

- All members of SAGE TF 3GPP stated that they were satisfied that the algorithms provide the high level of security for the f8 and f9 security functions required by 3GPP.

- All members of SAGE TF 3GPP approved release of the algorithms specifications and test data to 3GPP.

Publication of the algorithms specification

The SAGE TF 3GPP does not see from a security point of view any obstacles that would prevent publication of the f8 and f9 algorithms specifications.

In fact the SAGE TF 3GPP encourages such a publication because it would increase the public trust in the algorithm. It should be noted that though this is a significant advantage, publication could also initiate publications that try to discredit the security of the algorithms even if there is not much reality behind it. In some situations it might be needed to react on such publications.

11 Export control aspects

According to the cryptographic algorithm requirements in [1] it is the intention that:

"mobile stations should be free from restrictions on export or use, in order to allow the free circulation of 3G terminals, while network equipment which embody the algorithms may be expected to come under restrictions. It is however the intention that RNC and AuC which embody such algorithms should be exportable under the conditions of the Wassenaar Arrangement".

The SAGE Task Force made the following assumptions:

Mobile stations will not be controlled according to the Wassenaar arrangement, as long as they are "accompanying their user for the user's personal use". They would also be generally exempted from export control as being: "portable or mobile radiotelephones for civil use that are not capable of end-to-end encryption". The intended network wide encryption specified in the 3G architecture could possibly be debated but as it is only allowing network controlled key management it seems it would not qualify as true end-to-end encryption. The mobile stations are thus assumed to fulfil requirements according to [1], as long as the exporting countries abide by the Wassenaar rules.

Network equipment embodying algorithms should be expected to need export control licences according to the present Wassenaar arrangement (December 1998), very much like e.g. base stations for GSM have been and are export controlled today. The SAGE Task Force sees no reason to believe that any special problems should arise in this area

which could endanger the fulfilment of requirements for a wide international spread of 3G systems. The SAGE Task Force has, however, no possibilities to guarantee such a situation as the actual export licenses are handled individually by each country (or possibly internationally co-ordinated as by the European Union).

To some extent this topic was also discussed informally with a number of export control authorities and no adverse reactions to these interpretations were announced. It has also been noted by the SAGE Task Force that several countries have introduced more liberal rules than the Wassenaar arrangement indicates, especially in the area of so called mass market products, which the SAGE Task Force believes could even more alleviate the free movement of mobile stations.