3GPP TR 33.909 V1.0.0 (2000-12)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Report on the Evaluation of
3GPP Standard Confidentiality and Integrity Algorithms
(Release 1999)

Keywords

3GPP, Algorithm, Evaluation

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

Foreword

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

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.

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

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

1 Scope

This public report contains a detailed summary of the evaluation work performed during the design of the 3GPP Confidentiality and Integrity Algorithms. It contains all results and findings from this work and should be read as a supplement to the general report, ref. [6].

2 References

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

[1] 3GPP TS 33.102: "3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; 3G Security; Security Architecture".

[2] 3GPP TS 33.105: "3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; 3G Security; Cryptographic Algorithm Requirements".

[3] 3GPP TR 33.901: "3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; 3G Security; Criteria for cryptographic algorithm design process".

[4] ETSI/SAGE Specification: "Specification of the 3GPP Confidentiality and Integrity Algorithms; Part 1: f8 and f9 Specification". Version: draft; Date: 29th November 1999.

[5] ETSI/SAGE Specification: "Specification of the 3GPP Confidentiality and Integrity Algorithms; Part 2: KASUMI Specification". Version: draft; Date: 29th November 1999.

[6] ETSI/SAGE Report: "Report on the Design and Evaluation of the 3GPP Confidentiality and Integrity Algorithms". Version: draft; Date: 12th October 1999.

[7] ISO/IEC 9797-1:1999(E): "Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1".

[8] ISO/IEC 10116:1996: "Information technology – Security techniques – Modes of operation for an *n*-bit block cipher algorithm".

Additional references to external documents are provided in Annex A.

3 Abbreviations

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

AuC	Authentication Centre
CBC	Cipher Block Chaining
CK	Cipher Key
GF(q)	The finite field of q elements
3GPP	3rd Generation Partnership Project
f8	UMTS confidentiality (encryption) algorithm
f9	UMTS integrity algorithm
IK	Integrity Key
IV	Initialisation Vector
MAC	Message Authentication Code
OFB	Output Feedback Mode
RNC	Radio Network Controller
SAGE	Security Algorithms Group of Experts
SAGE TF 3GPP	SAGE Task Force for the design of the standard 3GPP Confidentiality and Integrity Algorithms
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
USIM	User Services Identity Module

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 a summary of the algorithm requirements;
- clause 7 consists of a brief presentation of the actual designs;
- clause 8 provides some background information on the chosen design;
- clause 9 gives an overview of the evaluation work carried out by SAGE TF 3GPP and other parties and the conclusions of the evaluations;
- Annex A includes a list of external references that are related to the results in this report.

5 Background to the design and evaluation work

The development of standardised 3GPP confidentiality and integrity algorithms was a major task that was to be conducted within a very short time period. There was a need for strong algorithms with a high degree of confidence, and the ETSI SAGE group decided on the following strategies for the work:

- Start with a cryptographic core based upon a public known block cipher that had already undergone extensive evaluation. Mitsubishi Electronic Corporation from Japan offered the MISTY1 algorithm for use in 3GPP. ETSI SAGE and 3GPP TSG SA3 agreed to select this design as the starting point for the work. MISTY1 was published at the Fast Software Encryption conference in 1997, ref. 0. The specifications of MISTY1 can be found at http://www.mitsubishi.com/ghp_japan/misty/index.htm.
- Expand the design team with experts outside the ETSI SAGE group.
- Invite interested manufacturers to participate in the design and evaluation process using their own cryptographic expertise.
- Perform additional evaluation by inviting several leading researchers for independent review of the final design.

Based upon the 3GPP requirements and the work conducted by the task force, a modified version of MISTY1 was developed and named KASUMI. KASUMI is the cryptographic engine of the 3GPP encryption and integrity algorithms f8 and f9.

6 Summary of algorithm requirements

The general security architecture for 3GPP is specified in ref. [1]. The complete set of security services needed is realised using a set of cryptographic functions identified in ref. [2]. Out of the full algorithm set there is a need for two algorithms fully standardised. These are:

- f8 – Confidentiality algorithm
- f9 – Integrity algorithm

The requirements for the f8 and f9 algorithms were specified in ref. [2] and can also be found in the Work Report ref. [6]. For this report we include some of the main requirements.

6.1 f8 – Confidentiality algorithm

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.

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).

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 f8 algorithm will be used to encrypt frames of variable length up to approximately 5000 bits.

The function f8 should be a symmetric synchronous stream cipher.

The length of the cipher key 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.

Additional input parameters: COUNT, BEARER, DIRECTION and 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.

6.2 f9 – Integrity algorithm

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

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

The function f9 shall be a MAC function.

The length of the integrity key 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.

Additional input parameters: COUNT, FRESH and LENGTH.

The algorithm shall output a 32-bit MAC.

6.3 Generic requirements for 3GPP cryptographic functions and algorithms

The functions should be designed with a view to their 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.

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 that 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 that RNC and AuC that embody such algorithms should be exportable under the conditions of the Wassenaar Arrangement [32].

7 3GPP confidentiality and integrity algorithms

The detailed specifications of the 3GPP algorithms are found in ref. [4] and ref. [5]. For this report we include a general overview of the design. The basic building block is the block cipher KASUMI, which is a Feistel block cipher with a block size of 64 bits and a 128-bit cipher key.

7.1 KASUMI

The structure of KASUMI is depicted in the following diagrams:

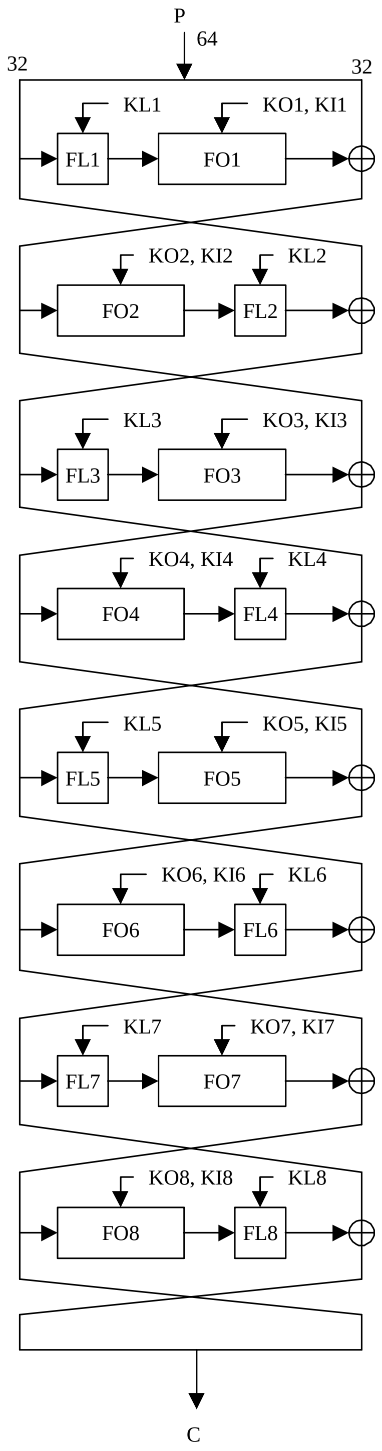

Fig. 1: KASUMI

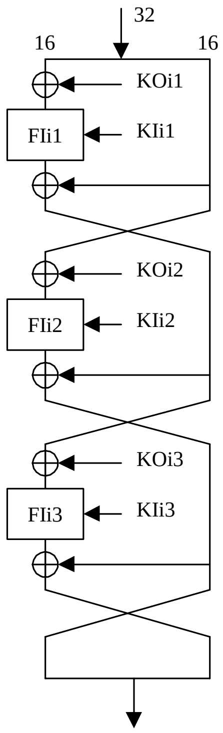

Fig.2: FO Function

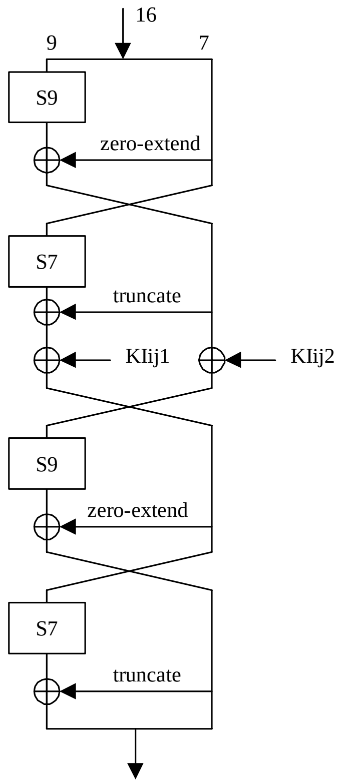

Fig.3: FI Function

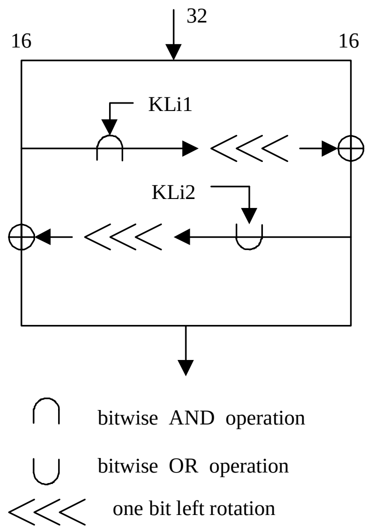

Fig.4: FL Function

KASUMI encrypts a 64-bit input by iterating a round function 8 times. The round function consists of the composition a 32-bit non-linear mixing function (FO) and a 32-bit linear mixing function (FL). The FO-function is again an iterated “ladder-design” consisting of 3 rounds of a 16-bit non-linear mixing function FI. In turn, FI is again defined as a 4-round structure using non-linear look-up tables S7 and S9. All functions involved will mix the data input with key material. See ref. [5] for details on the specification of S-boxes and generation of round keys.

7.2 Confidentiality function f8

The stream cipher f8 used for encryption of data frames in 3GPP is constructed from KASUMI in a variant of the standard Output Feedback Mode (OFB) ref. 0, with 64-bit feedback. The construction is depicted in the following diagram:

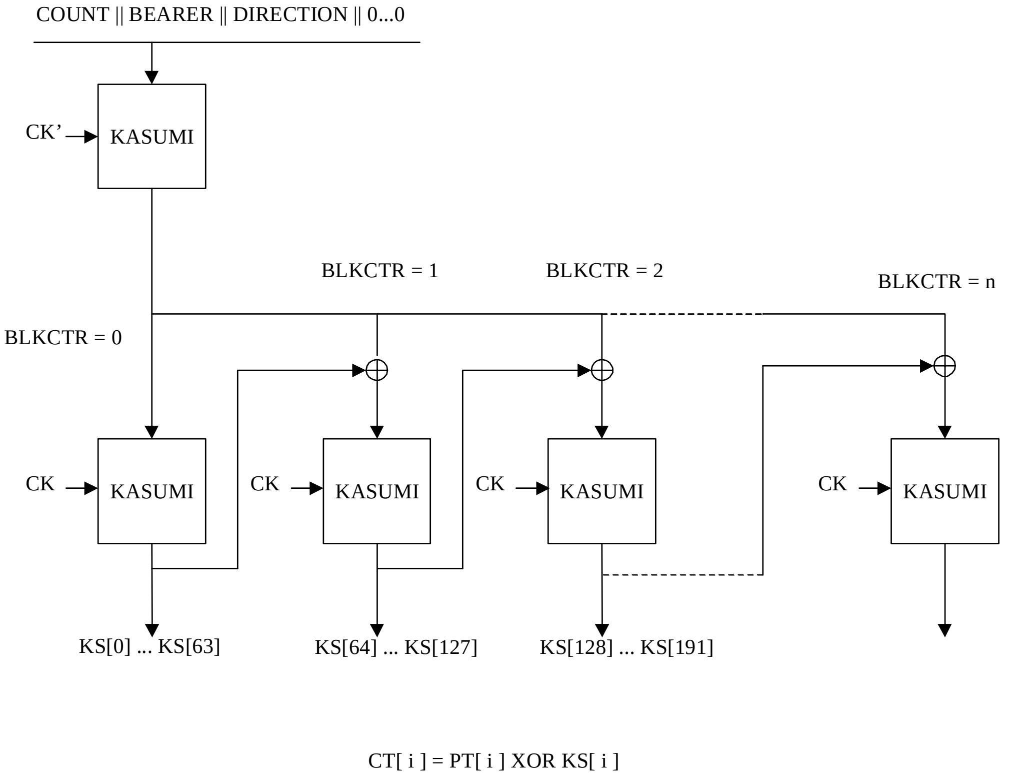

Figure 5

During a pre-computation phase, the system parameters COUNT, BEARER and DIRECTION are padded to become a full length datablock and KASUMI encrypted with a derived key CK’. The derived key CK’ is the exclusive or of the original cipher key CK and a fixed mask. The output of this process is a 64-bits register value A, which is part of the input in each subsequent KASUMI computation.

Subsequent blocks (64 bits) of keystream are then generated by running KASUMI in output feedback mode with additional input of A and the block counter (BLKCTR) to the feedback. The cipher text is then produced as the exclusive or of the keystream bits and the plaintext bits.

7.3 Integrity function f9

The integrity algorithm f9 computes a 32-bit Message Authentication Code (MAC) on an input message under an integrity key IK. The message may be between 1 and 5000 bits in length. The structure of f9 is depicted in the following diagram:

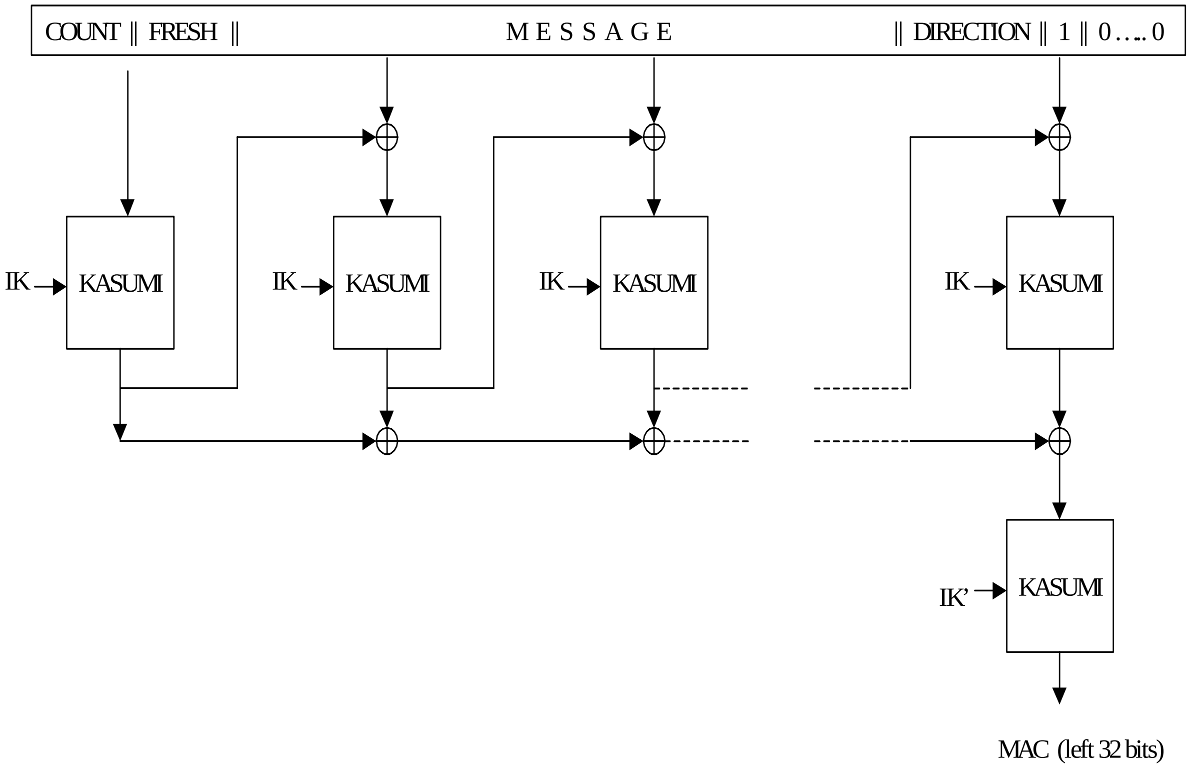

Figure 6: The integrity function f9

f9 is a variant of the standard CBC MAC as defined in ISO 9797 (ref. [7]). In this structure the exclusive or of the outputs from all block computations is xored to the input of the final KASUMI computation (output transformation). The MAC on the input message consists of the left half of the output from this last computation.

8 Rationale for the chosen design

8.1 General Comments

The essential design goals for the 3GPP confidentiality and integrity algorithms were that the algorithms should:

- provide a high level of security within the 3GPP context;
- meet their implementation requirements — in particular, allow a low power, low gate count implementation in hardware.

The designers have therefore deliberately avoided over-designing the algorithm. They wanted the algorithms to be secure against all practical attacks, and carefully decided not to over-complicate them just to provide a very high security margin against unrealistic theoretical attacks.

The following types of attack against the underlying block cipher KASUMI were particularly considered:

- linear cryptanalysis;
- differential cryptanalysis, and variants such as impossible differentials, “miss in the middle”, etc;

- higher order differential cryptanalysis and interpolation, including probabilistic higher order analysis;
- identifying any classes of weak keys.

No weak keys were found. There are chosen plaintext and/or related key attacks against KASUMI reduced to 5 rounds, see section 9.2. We believe that with further analysis it might be possible to extend some attacks to 6 rounds, but not to the full 8 round KASUMI. In any case, the more powerful attacks do not translate to practical attacks against the f8 and f9 algorithms in the 3GPP context.

There are several obvious ways to increase the security margin offered by KASUMI. These include:

- increasing the number of rounds;
- adding a fourth round to the FO function;
- making the key schedule more complicated.

All of these were considered, and rejected as adding complexity for no practical gain.

Attacks against the f8 and f9 constructions were also considered. The f8 construction is a good example of the pragmatic approach to the design. Given a very long sequence of keystream (of order 2^{38} bits), it would be possible to identify a small amount of structure in the keystream, which could be classified as an attack (or at least an imperfection); but in the 3GPP context, frames of keystream will be no more than 5000 bits long, so the designers saw no need to protect further against this kind of attack.

8.2 Design Policy of MISTY1

The 3GPP crypto engine KASUMI is based on the block cipher MISTY1, ref. [19], which was designed according to the following three principles:

- MISTY should have a numerical basis for its security;
- MISTY should be reasonably fast in software on any processor;
- MISTY should be sufficiently fast in hardware implementation.

The algorithm was designed to be provably secure against differential and linear cryptanalysis. This results from building the algorithm according to provable constructions from smaller components with known resistance against these two types of attacks. The Feistel structure of MISTY1 is recursively repeated in the smaller round function FO and in the kernel FI.

The unequal division of FI is due to the fact that bijective functions of odd size are generally better than those of even size from the viewpoint of provable security against linear and differential cryptanalysis.

In selecting the S-boxes $S7$ and $S9$, the following criteria were adopted:

- Their average differential/linear probability must be minimal;
- Their delay time in hardware is as short of possible;
- Their algebraic degree is high, if possible.

The resulting functions were found by searching for functions of the form $A(x^t)$ over $GF(2^7)$ and $GF(2^9)$, where A is a bijective linear transformation. The non-linear degree of $S7$ is 3 and the non-linear degree of $S9$ is 2.

For the purpose of avoiding possible attacks other than differential and linear cryptanalysis, the design of MISTY1 was supplemented with the simple and fast function FL. This function is linear for a fixed key, but has a variable form depending on the key value.

The key scheduling part of MISTY1 was designed according to the following principles:

- The size of the key is 128 bits;
- The size of the subkey is 256 bits;
- Every round is affected by all key bits;

- As many subkey bits as possible affect every round.

8.3 Changes from MISTY1 to KASUMI

This section summarises the changes that have been done to MISTY1 during the design of KASUMI.

8.3.1 Data Encryption Part

a) Changing the location of the FL functions

This makes hardware simpler; (but a bit slower - this drawback is recovered with other changes. Note that this structure does not block parallel computation of two FI functions).

b) Removing the subkey KO_{i4} in the FO function.

This makes hardware simpler and faster; as the FO function now has a simple repetitive structure.

c) Adding rotate shift functions in the FL function.

It is assumed that this makes cryptanalysis harder and has no negative impact on hardware size and speed.

d) Changing of the substitution table *S*7.

This is not a significant change, and is in fact equivalent to just rearranging the bit order before and after the original *S*7. We have not found a better table from the viewpoint of hardware implementation.

e) Changing of the substitution table *S*9.

This makes hardware smaller (and possibly faster). The total number of “terms" of the new *S*9 in its algebraic normal form is smaller than that of the original *S*9. We searched all polynomials and normal bases, all powers whose hamming weight is two, and all linear combinations of t_j's for shorter y_i's (see [5]), where the length of y_i is defined as the number of terms (except a constant value) in its algebraic normal form. For the new *S*9, the average length of yi's is 11.2, while for the original version it is 11.7.

f) Adding another *S*7 in the FI function

This makes the security level significantly higher but hardware bigger. We expect that this increase will be compensated with the reduction of the key scheduling part. Note that the penalty on hardware speed is not particularly significant because *S*9 and *S*7 can be performed in parallel.

8.3.2 Key Scheduling Part

a) Removing all FI functions in the key scheduling part.

This makes hardware smaller and/or reduces key set-up time. We expect that related key attacks do not work for this structure.

b) Adding the constant values Ci and rotate shift operations.

This avoids using the same subkey values in different rounds.

9 Algorithm evaluation

9.1 Evaluation criteria

The evaluation work performed by the extended task force was divided into three different parts. The main investigations in each part are described in this section.

9.1.1 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 are investigated.

9.1.2 Analysis of KASUMI as a generic 64-bits blockcipher

This represents the main conponent 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 are investigated.

One can (informally) describe as an attack of a 64-bit 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 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 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, ref. [11], *boomerang attacks*, ref. [31], *truncated differentials*, ref. [14], 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**: finding 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*, ref. [18], *higher order cryptanalysis*, ref. [17], and other statistical cryptanalysis methods.
- **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.

9.1.3 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.

9.2 Mathematical analysis of KASUMI

9.2.1 Properties of components

Each functional component of KASUMI has been carefully studied to reveal any weakness that could be used as a basis for an attack on the entire algorithm. The following are the main results from this work.

9.2.1.1 FL function

The FL function is a linear function, and the security of the algorithm is not meant to depend on this function. Its main purpose is to be a low cost additional scrambling, making individual bits harder to track through the rounds.

The FL function has the property that for any key KL, an input of $0^{16}1^{16}$ always gives an output of 1^{32}. Hence for some round inputs, some of the key bits in KL can be changed without having any effect on the output of that round. This property can be used to guarantee a zero difference at the end of the first round, thus effectively removing the first round. More generally, small changes to the input to FL only make small output changes, and this can be useful going either forwards or backwards through FL.

The fixed point is used in some of the differential attacks mentioned later, but no attack exploiting this property that extends beyond 5 rounds of KASUMI has been found.

9.2.1.2 FI function

This is the basic randomising function of KASUMI with 16 bits input and 16 bits output. It is again composed of a four-round structure using two non-linear substitution boxes S7 and S9. Using theorem 4 of ref. [19], we can show that the average linear and differential probability of FI is less than $(2^{-9+1})(2^{-7+1}) = 2^{-14}$ assuming uniform distribution of the subkeys in use. S7 and S9 have been designed in a way that avoids linear structures in FI. This fact has been confirmed by statistical testing.

The Walsh spectra of the outputs of FI for several keys have also been computed. As expected, they show clear peaks. The spectrum of the FI bits seen as a function of the key behaves in a very regular fashion. There are 256 peaks beside the zero frequency component, all of absolute value 2048. There are always (for different *x* entries of FI(*x*, *k*)) “frequencies” between index 1088 and 1344. The position of the peaks does not change.

The explanation for this behaviour is the low degree of the S-boxes together with the fact that the round keys KI1 and KI2, which are fed into the function, only go through one S-box each. From the FI-structure the non-linear order of the output bits can easily be calculated as function of some or all input bits. Such orders are used for determining the resistance of the cipher against higher order differential attacks.FO function.

The FO function constitutes the non-linear part of the KASUMI round function. Again using theorem 4 of ref. [19], we can show that the average linear and differential probability of FO is less than 2^{-28} assuming uniform distribution of the subkeys in use. For any fixed key, FO is a permutation of 32-bits blocks, but due to its 3-round structure it can be distinguished from a randomly chosen permutation using four chosen plaintexts.

It was considered to improve the diffusion properties of the FO by adding a fourth round, like the FI structure. On the cost of adding complexity and power consumption this could improve to the general security margins of KASUMI. However, there are no indications that the properties of a three round FO can be used in an attack of the full 8 round KASUMI.

9.2.1.3 The S7 box

The S7 box in KASUMI is essentially the same as S7 in MISTY1 (see [19]); the KASUMI S7 was made by rearranging the bit order before and after the original S7. The S7 box is specially designed to be easy to implement in hardware using combinational logic, and as a consequence the non-linear order is 3. The algebraic normal form of this function can be found in ref. [5]. Some of the properties of this box are described below.

9.2.1.3.1 Kasami exponent

The substitution box S7 is a linear transform of the monomial x^{81} defined over GF(2^7), which has optimal nonlinearity properties. The exponent 81 belongs to the set of exponents proved by Kasami in 1971 to give the optimal nonlinearity properties (see [12]).

Kasami's result says that for $n = 2m + 1$, $2 \le k \le m$, and $\gcd(k,m) = 1$, the power function x^d, where $d = 2^{2k} - 2^k + 1 \pmod{2^n - 1}$, is maximally (i.e., almost perfect) non-linear. We say that exponent d' is equivalent to d if there is t such that $d' = 2^t d$. Clearly, then the power functions x^d and $x^{d'}$ have the same nonlinearity properties.

We see that $81 = 2^6 + 2^4 + 1 = 2^4 (2^4 - 2^2 + 1) \pmod{2^7 - 1}$, and that, for n = 7, $13 = 2^4 - 2^2 + 1$ is a Kasami exponent with k = 2.

9.2.1.3.2 Probabilistic approximation

In probabilistic cryptanalysis one tries to approximate a functional block by low degree functions. Attacks based on this approach have shown to be able to break block ciphers that were proven to be resistant against linear/differential attacks, see [13]. To assess if this approach can be successful we considered the following:

Input: n input/output pairs $\{(x_i, y_i)\}$, $x_i, y_i \in$ GF(q) and chosen constants d and t

Output: All polynomials $f(x, y) = y - p(x)$, $\deg p \le d$ such that $|\{x_i; f(x_i, y_i) = 0\}| \ge t$ (if they exist)

To find the polynomials *f* one can use Sudan's algorithm described in [27] with the modification described in [10]. We applied the Sudan algorithm to S7 to see whether the low degree expression over GF(2) gives rise to probabilistic approximations of low degree over GF(2^7). For several trials with random selection of the *n* points and different choice of parameters no approximation of degree 6 or lower was found except when the number of points was extremely small.

Thus it seems impossible to get a good probabilistic approximation over GF(2^7).

9.2.1.4 Cycle structure

The cycle structure of the S7 permutation was found to be:

Cycle Length	No of cycles
92	1
22	1
13	1
1	1

Thus S7 has one fix-point given by S7(27) = 27, but no obvious deficiencies can be found from the cycle structure.

9.2.1.5 The S9 box

The S9 box is different from the S9 box in MISTY1 [19], but it has been constructed in much the same way. That is, it is easy to implement in hardware (actually easier than the original S9), and has non-linear order 2. The algebraic normal form of this function can be found in ref. [5]. S9 can been seen as a composition of the power function x # x^5 and a linear output transformation defined over $\mathrm{GF}(2)^9$, it is known that it achieves almost perfect non-linearity.

9.2.1.5.1 Linear structures

Let $s : GF(2^n) \rightarrow GF(2^n)$ be an almost perfect non-linear permutation, the individual bits of whose output are quadratic functions of the input bits. Let

$$f_{\underline{u}}(\underline{x}) = s(\underline{x}) \bullet \underline{u}$$

be any non-zero linear combination of the output bits of s. Then there is precisely one linear structure of f_u, i.e. one non-zero vector $\underline{w}$ such that $f_{\underline{u}}(\underline{x}) \oplus f_{\underline{u}}(\underline{x} \oplus \underline{w})$ is constant. We can call this a linear structure $(\underline{u}, \underline{w})$ of the permutation s. See reference [23]. So our table S9 is bound to have 511 linear structures $(\underline{u}, \underline{w})$.

However, the S9 table has been constructed in such a way that this does not lead to any linear structures in FI, and hence in FO.

9.2.1.6 Cycle structure

The cycle structure of the S9 permutation was found to be:

Cycle Length	No of cycles
275	1
121	1
74	1
26	1
12	1
2	1
1	2

This means that the cycle structure does not deviate from the expected structure for a random permutation.

9.2.1.7 Key schedule

The key schedule of KASUMI is very simple, but this fact has not been found to constitute any real weakness, and there seems to be no gain in practice by making it more complicated. Each of the 128 bits of secret key is used one and only once in every round. They are used in different ways in different rounds, and also at different parts within those rounds, and at times the values are altered using key modifications constants.

Due to the use of the constants C1 to C8 in the key schedule, there is no fixed recurrence relation between consecutive round keys. This property is required to prevent chosen plaintext attacks that are faster than exhaustive search. Further, there exists no equivalent, more compact representation of the expanded key.

Even if regularity and symmetry in the key scheduling do not introduce weaknesses in the algorithm, care should be taken such that shorter keys e.g. 64 bit keys are not extended to a full-length key in a very symmetric way. Just padding with zeroes could give some advantage to an attacker and should not be recommended.

In his analysis of MISTY1 [19] Matsui shows that if the subkey bits are independent, the average differential and linear probabilities are less than 2^{-56}. Some concern has been expressed that with the simple key schedule in KASUMI, the assumption of subkey independence might be too optimistic. However, we have no indications in this direction. (See also section 9.2.4).

9.2.2 Differential cryptanalysis

From its construction it is clear that, provided that subkeys are independent, three rounds of KASUMI have no differential or linear characteristics with probability larger than 2^{-56}. It should be noted that the upper bounds on FI, see 9.2.1.2, is tight. It is possible to find differential characteristics for FI with probability 2^{-14}. It is also important to note that the differential effect of FL is low.

In this section we review some of the differential attacks that have been found on reduced versions of KASUMI.

9.2.2.1 A differential chosen plaintext attack

This section describes a chosen plaintext attack on 5 rounds of KASUMI that can be used to recover the key. The attack requires roughly 2^{38} chosen plaintexts, and 2^{80} small operations.

Denote by Xi (i=0..9) the 32-bits words encountered at the inputs or outputs of the various rounds. Thus P=(X1,X0), C=(X9,X8), and round i is summarised by the relation Xi+1 = Fi [Xi] $\oplus$ Xi-1 (i=1 to 8), where Fi=FOi $\lfloor$ FLi if i is odd, and Fi = FLi $\lfloor$ FOi if i is even. Then X4 $\oplus$ X0 is a one to one function of X0. This means that when performing 2^{32} encryptions with the same X1, but different X0s, no collision will occur among the obtained X4 $\oplus$ X0 values.

The attack uses the following consequence of this property:

(P) Denote by SUM(K, X1) the 32-bits XOR of the 2^{32} X4 values associated with the 2^{32} possible X0 values (thus given a fixed X1 value, we consider the encryption, under the same unknown key K, of the 2^{32} (X1, X0) plaintexts which left half equals X1). The following relation holds:

$\forall$ K $\forall$X1 SUM(K, X1) = 0

The attack requires a number of (X1, X0) chosen plaintexts that lie within a small factor of 2^{32} - say 2^{38} plaintexts- and the corresponding (X6, X5) ciphertexts. The plaintexts are derived from a small set of say 2^{6} X1 fixed arbitrary values. For each of the X1 values, the 2^{32} (X1, X0) plaintexts in which the left half equals X1 are considered.

The attack uses property (P) to test parts of the 5th round key. When the 82-bits (KL5, KO51, KI512, KO52, KI522) value have been found, the remaining 42 unknown bits of K can be determined by exhaustive search.

Using linearisation methods similar to those described in [29], it might be possible to extend this attack to 6 rounds, but not to the full 8 rounds of KASUMI.

Attacks based on differentials with low Hamming weights have also been considered. For FL is it true that an input difference with low Hamming weight evolves to an output difference with a low Hamming weight. However, the same property does not hold for FO and no substantial attack on KASUMI could be found.

9.2.2.2 Differential related key attacks

Though related key attacks seem to be no threat in the 3GPP context, we have considered this type of attack, and concluded that it is possible to perform differential related key attacks on four and five rounds of KASUMI. The four round attack requires the encryptions of approximately 2^{9} chosen plaintext pairs X and X^* under keys K and K^* respectively, where K and K^* differ in only one bit. The average complexity of this attack is approximately 2^{41}. The five round attack, which is an extension of the four round attack, requires the encryptions of on average $3{\cdot}2^{17}$ chosen plaintext pairs, and has an average complexity of approximately 2^{36}.

The requirement specification [2] states that if the key shall need to be shorter than 128 bits, the least significant bits shall be set to zero. If the key is reduced to only 64 bits, that is, $K_5 = \cdots = K_8 = 0^{16}$, the algorithm is vulnerable to a five round related key attack that only needs about 10 plaintexts encrypted under two keys and has a complexity of roughly 2^{25}.

These attacks all rely on essentially the same differential, which predicts differences at the end of the third round with a probability of one half. This differential arises since the subkey K_3 appears early in all of the first three rounds. Changing the order in which the subkeys are fed into the rounds, for example, could destroy this differential, but such a change could create other differentials in other rounds.

In any case no attack of this type has been found that extends beyond 5 rounds of KASUMI, and, as stated, related key attacks are no threat in the 3GPP context.

9.2.2.3 Impossible differentials

In the FI function there are no impossible differentials, because of its four-round structure. In the three round FO function, however, several impossible differentials occur since the round function FI is bijective. These lead to impossible differentials over 2 and 3 rounds of KASUMI without the FL function.

The FL functions seem to destroy most of these impossible differentials, or more precisely, make their existence key dependent. We were not able to derive any impossible differentials for the true KASUMI from those known to exist for the FO function.

Hence we are not aware of other impossible key-independent differentials for the KASUMI cipher, than the well-known five-round impossible differential of the form:

$$(0, A) \rightarrow (A, 0) \rightarrow (*, A) \rightarrow (A, *) \rightarrow (0, A) \rightarrow (A, 0)$$

where A is a non-zero 32-bit block, 0 is a 32-bit block of all zeros and each occurrence of $*$ can be replaced by any (possibly different) non-zero block.

This differential can however be used to distinguish 5 round KASUMI from a truly random function:

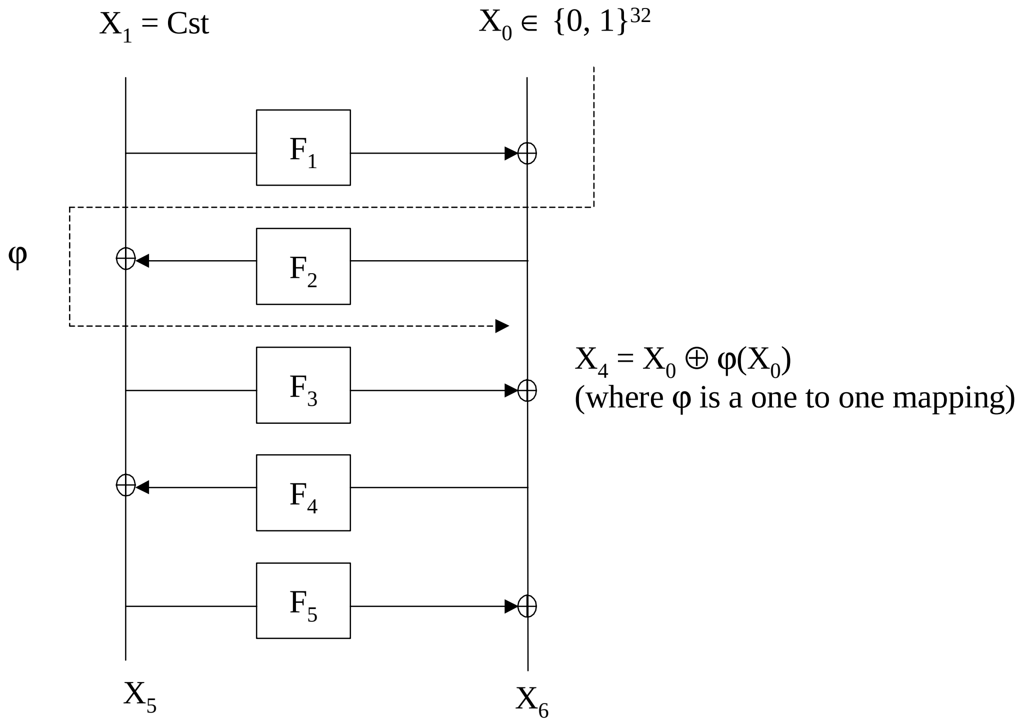

Figure 7: The 5 first rounds of KASUMI

Consider a $((X_1,X_0), (X'_1,X'_0))$ pair of plaintext blocks, and denote by ΔX_i the resulting intermediate difference $X_i \oplus X'_i$. Then we have the following distinguishing property:

If $\Delta X_1 = 0$ and $\Delta X_0 \neq 0$, then one cannot have simultaneously $\Delta X_5 = 0$ and $\Delta X_6 = \Delta X_0$.

This is because if $\Delta X_1=0$ and $\Delta X_5 = 0$, then $\Delta X_6 = \Delta X_0 \oplus \Delta\varphi(X_0)$. Thus if in addition $\Delta X_6 = \Delta X_0$, then $\Delta\varphi(X_0) = 0$, which contradicts $\Delta X_0 \neq 0$ since φ is one to one. The consequence is that KASUMI restricted to 5 rounds can be distinguished from a random permutation with a probability close to 1 with slightly more than 2^{32} plaintext/ciphertext pairs. Lucks' attack on DEAL is based upon the same impossible differential, see [24].

An attack on KASUMI reduced to 6 rounds has been found that requires 2^{55} chosen plaintexts and computation of approximately 2^{119} FI values. Another attack against 6 rounds of KASUMI has been found requiring $2^{53.3}$ chosen plaintexts with a complexity of the order of 2^{100} encryptions. Both attacks exploit impossible differentials and the structure of the FO function.

No similar attack on the full 8 rounds of KASUMI has been found, and in the 3GPP context these attacks are not applicable.

9.2.3 Truncated differentials

The best way that has been found to exploit truncated differentials for KASUMI leads to an attack on 3 or 4 rounds of KASUMI without the FL function. This attack uses the fact that the function FO restricted to the 16 leftmost input bits, is bijective onto the leftmost 16 bits in the output.

3 rounds can be broken using about 2^{35} plaintext pairs derived from 2^{18} chosen plaintexts. The 4 round attack requires 2^{48} chosen plaintexts. The FL function will complicate the attack, and in any case, KASUMI with 5 rounds or more is secure against this attack.

9.2.4 Linear cryptanalysis

The validity of the proofs of security given by Matsui in [19] has been examined. That is, how average is the behaviour of fixed keys with respect to linear approximations over the FI function. Mathematical calculations using the Walsh-Hadamard transform and experimental calculations were carried out independently and reached the same conclusions.

LP^{FI} was estimated to be on average smaller than 2^{-14} for any linear hull over FI, but there are specific key values and linear hulls for which $LP^{FI} \approx 2^{-12}$. Of course there will also be key values for which the actual bias is much less than the average case. The maximal amounts of correlation are not high enough to make it possible to chain them to a useful linear approximation path over rounds of KASUMI. For construction of overall approximations one needs to consider all possible paths, and not only the ones which give large biases (correlations).

One attack on five rounds of KASUMI might be possible, but it would require a work effort of at least 2^{95}, around 2^{58} known plaintexts and only be applicable to a fraction of 2^{-3} of the key space. A variant may potentially reduce the work effort to 2^{93} and require around 2^{49} known plaintexts, but will only be applicable to a fraction of 2^{-41} of the key space.

We conclude that, for the full 8 round KASUMI, all keys of the FI function behave pretty much like an average key with respect to the studied linear approximation relations.

9.2.5 Higher order differential attacks

Quite a lot of analysis has been conducted in Japan concerning the strength of the Misty algorithms. Tanaka et al. shows in [29] that 5 round Misty1 without the FL function can be attacked using 1,408 chosen plaintexts, with a method using 6th and 7th order differentials.

It can be shown that the differential property leading to this attack is actually due to the choice of the *S*7 box. Further, it can be shown that it is actually not possible to find an *S*7 box coming from a mapping $x \# x^{e_3}$ with an exponent e_3 of Hamming weight 3, that is at the same time an optimum from the points of view of average differential/linear probabilities and of the 7th order differential property. Finally, the product of any two output bits from S7 will have an algebraic degree bounded by 5.

However, we do not believe that this 7th order differential property still holds for KASUMI, due to the modification of the FI function. Further, we are convinced that traditional attacks based on higher order differentials will work for at most 5 rounds of KASUMI, and no other variants have been found that work for more than 5 rounds of KASUMI.

In [28] Sugita shows the relation between inputs and outputs of 6 rounds of a “Misty-like” transformation, and proves it is not a locally random function. The relation is used in a higher order differential attack to guess the key of 5 round Misty1 without the FL function.

9.3 Implementation attacks

KASUMI has also been analysed with respect to differential attacks like *timing attacks, simple power analysis and differential power analysis.* This investigation did not reveal any properties of KASUMI that would make it particularly vulnerable to these type of attacks. Specifically KASUMI has a favourable key scheduling with respect to power attack methods that try to derive information about the Hamming weight of subkey bytes. The restricted use of KASUMI in the 3GPP environment will also reduce the possibilities for such attacks. In an application where an attacker can do measurements of time of execution and/or power consumption, specific care should be taken to guarantee resistance against implementation attacks.

9.4 Analysis of f8 and f9

9.4.1 Supporting arguments for the f8 construction

The final f8 construction uses a pre-computation of a pre-whitening constant:

$$W = KASUMI_{CK\ XOR\ KM}(Count||Bearer||Direction||0..0)$$

The two main advantages expected from such a «pre-whitening» are the following:

1) **Protection against chosen plaintext attacks:** if the IV = Count||Bearer ||Direction||0..0 block is used directly in the OFB chain, then, in case of a known plaintext, the input and output of each KASUMI operation are known. With pre-whitening the plaintext inputs to the KASUMI are no longer known.

2) **Protection against collision attacks:** analysis shows that it would in principle be possible to distinguish the pseudo random function generator associated with an f8 construction without the pre-whitening from a truly random generator, based on the observation of say 2^{33} keystream blocks. This does not seem to be the case with the actual f8 construction. As a matter of fact:

   - for distinct initial values IV and IV', the pre-whitening constants W and W' differ, and it becomes difficult to predict that certain keystream blocks associated with IV and IV' are equal (on the other hand, the observation of two equal keystream blocks associated with two distinct IV values provides an adversary with the W XOR W' value, but this does not represent by itself a distinguishing information : this allows to predict which other pairs of blocks associated with IV and IV' are equal, but since there are less than 80 blocks in both keystreams, the probability of such a "distinguishing event" is less than $80^2/2.2^{64} \approx 2^{-52}$).

   - given a fixed initial value IV, the following distinguishing property holds: collisions on keystream blocks are predictable, but the probability that such collisions occur among the at most 80 blocks of the keystream associated with IV is less than $80^2/2.2^{64} \approx 2^{-52}$. So even if an adversary is provided with the keystream sequences associated with all possible count values, the distinguishing probability remains low.

9.4.2 On the Construction of f9

If a regular CBC-MAC mode had been chosen for the f9 algorithm, the internal state fed forward from block to block would have been only 64 bits long. In this case a 2^{33}-message birthday attack would be likely to yield an internal state coincidence. Having identified a pair (m_i, m_j) for which such a coincidence occur, you can always be sure that $m_i||x$ and $m_j||x$ have the same MAC for any extension x. In other words, if you can obtain the MAC for $m_i||x$, then you can forge the MAC for $m_j||x$.

This attack would be unrealistic in the 3GPP context, but nevertheless the current f9 construction has been chosen over the regular CBC-MAC mode because it provides a 128-bit internal state at almost no extra cost. The f9 construction prevents the 2^{33}-message birthday attack, seemingly without introducing any other weaknesses. The straightforward birthday attack on this construction requires 2^65 chosen-texts, which is completely out of reach. A variation on the birthday attack found by Knudsen and Mitchell [14a] requires approximately 2^{48} chosen-texts, which is still considerably more than for the regular CBC-MAC mode.

The following observations can be made on f9; none of these seem to present any security weakness.

- A change in a single block will no longer change the MAC with probability one (except for the last block), This property is satisfied by standard CBC-MAC, but not by f9.

- For every value of the x of the chaining variable, there exists an input block y such that the output again is x. Note that both x and y are completely unknown, and both values depend on the value of the integrity key. Then inserting the block y an even number of times will not affect the MAC value.

- As a special case of the previous fact, if $x = 0$, which is an event with probability 2^{-64} (that cannot be detected easily by an opponent), inserting y (which again is hard to find) an arbitrary number of times will not affect the MAC value.

9.5 Statistical evaluation

The objective of the statistical evaluation is to make sure that the block cipher KASUMI, the confidentiality function f8, and the integrity function f9 behave like random functions, i.e., that the test results do not indicate a deviation from random behaviour.

9.5.1 Criteria for statistical evaluation

To assess the statistical behaviour of the block cipher KASUMI, we started with the statistical evaluation of the building blocks of KASUMI. KASUMI consists of different non-linear functions: the two S-boxes S7 and S9 and the two functions FI and FO (see section 7.1). The FL function was not considered in the statistical evaluation.

The following statistical tests or calculations were performed on the two S-boxes:

- Linear approximation test ([25])
- Test on linear structures ([20])
- Cycles of the S-boxes
- Differential tests
- Dependence tests (Avalanche effect)

The two functions FI and FO were tested by the test on linear structures and the dependence test.

After looking at the building blocks, the block cipher KASUMI itself was tested. We used the dependence test to see if KASUMI satisfies the plaintext-ciphertext Avalanche effect and the key-ciphertext Avalanche effect. The Avalanche effect demands that about 32 bits of the output block shall change if one bit of the 64-bit input block is toggled if the key is fixed, or if one bit of the 128-bit key is toggled provided the same input block is used. We performed the dependence test on KASUMI reduced to two rounds, reduced to four rounds and on the full round KASUMI.

To check how good KASUMI destroys redundancy in the input data, we generated a sequence of 16384 blocks of 64 bits by consecutive applications of the block cipher algorithm in ECB-mode, where between two encryption operations the input block is increased by one, starting with the all-zero block. The key was randomly chosen and the same for all calls of KASUMI. For the first sequence the output blocks were concatenated to a sequence of 1,048,576 bits. For the second sequence we built 64 sequences of 16384 bits each out of the i^{th} bits of the 16384 blocks, i.e. one sequence consisting of the first bits of all 16384 blocks, one sequence consisting of the second bits of all blocks, …, one sequence consisting of the 64th bits of all blocks. These 64 sequences were then concatenated again to a sequence of 1,048,576 bits.

The following statistical tests (stream cipher tests) were applied on the two sequences (for a description of most of the tests see for example [15]):

- 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

- Maximum-order-complexity test

The f8 function 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 stream cipher tests on a lengthy sequence produced by f8 were performed.

Two sequences were produced: a long keystream sequence and a sequence built from concatenation of small keystream sequences. On both sequences the stream cipher tests listed above were performed.

For the long keystream sequence CK, COUNT, BEARER and DIRECTION were chosen randomly but fixed and BLKCNT was incremented by one for each block cipher encryption, starting with zero. The iterated block cipher encryption was performed 32,768 times, such that a sequence of 2,097,152 bits was produced. This covers 15 out of the16 bits of BLKCNT.

In reality algorithm f8 will be used to produce many small keystream blocks which are used to encrypt different physical layer frames. Each frame will be encrypted with a new keystream block produced by f8. Thus the whole data stream is encrypted by a concatenation of small separately produced keystream blocks. The keystream blocks will have a length between 1 and 5000 bits. COUNT, BEARER, DIRECTION, and CK were randomly chosen. The keystream was then generated according to the construction of the f8 function with randomly chosen LENGTH value. So the keystream as a whole is a concatenation of small keystream sequences.

The stream cipher tests mentioned above were applied to the sequences generated by the output of the f8 function.

For the integrity mode f9 it is essential that the MAC depends on every bit of the input. This was proved by the dependence test. Two test scenarios were considered: in the first scenario one has a fixed message and a fixed initialisation vector (IV, consisting of COUNT, FRESH and DIRECTION) and toggles the bits of the integrity key IK. In the second scenario IK is fixed and one toggles the bits of the IV and the message.

The first test proves that the 32-bit MAC depends on every bit of the Integrity Key IK (key-ciphertext Avalanche effect). We randomly chose a fixed message of 100 bits, thus the message is split into two blocks, and a 65 bit IV (COUNT, FRESH and DIRECTION). Because of the padding, the input as a whole was 192 bits (three blocks). The dependence test was performed where the bits of the integrity key IK were toggled.

The second test proves that the 32-bit MAC depends on every bit of the input, i.e. the IV (COUNT, FRESH and DIRECTION) and the message itself (plaintext-ciphertext Avalanche effect). For this test we chose a fixed integrity key IK. We used a 100-bit message, thus the message was split into two blocks, and the 65-bit IV as input. Because of the padding, the input as a whole was 192 bits (three blocks), but the bits added by the padding were not toggled.

9.5.2 Results from statistical test

The two S-boxes S7 and S9 are Almost Perfect Non-linear (APN) bijective Boolean Mappings. It is known from the literature (e.g. [12] and [22]) that those functions have specific properties. Some results from the calculations and tests made are due to the construction of the S-boxes. The statistical tests confirmed the design principles of the actual constructions.

The linear approximation test showed that in the case of S7 the maximal Hamming distance of each linear combination of the output components of S7 from the set of affine functions is equal to 64 + 8 = 72, i.e. each linear combination of the output components can be approximated by at least one affine function up to 64 + 8 = 72 values. For S9 the value for the Hamming distance described above is equal to 256 + 16 = 272. That is because the Walsh transform of each linear combination of the output components of the S7 and S9 mappings are three-valued (see [12]).

S7 has no linear factors. But for each linear combination of the output components of S9 one can find one linear factor. That is due to the fact that the component functions of S9 are quadratic (see [22]), i.e. the algebraic normal form of the component function has quadratic terms at the most.

Concerning the cycle structure, S7 and S9 have no obvious deficiencies, e.g. a lot of transpositions.

S7 and S9 are not random S-boxes. The dependence test showed that each output bit of both mappings is dependent on every input bit. But for S9 there are output bits which always change when one input bit is toggled. This is because of the linear structures of S9. S7 satisfies the Avalanche effect, S9 does not.

For the FI and the FO functions no linear structures were found. The dependence test showed that each output bit of both functions is dependent on every input bit. Both functions satisfy the Avalanche effect. But a closer look at the FI function shows that it doesn’t behave like a random function according to the dependence test.

KASUMI reduced to four rounds already satisfies the key-ciphertext and the plaintext-ciphertext Avalanche effect.

The two sequences generated to verify that KASUMI destroys redundancy in the input passed all stream cipher tests, i.e. there is no indication that these sequences deviate from random behaviour. The same holds also for the two sequences (long keystream sequence, concatenation of small keystream sequences) generated by the f8 function.

The f9 mode satisfies the key-ciphertext and the plaintext-ciphertext Avalanche effect.

An additional dependency test was performed on the double encryption with two related keys in the beginning of f8. This test did not show any statistical weaknesses in this construction.

9.6 Results from independent evaluation

This section contains the basic findings from the group of three independent evaluators. For each group their summary includes the conclusions of their work and optionally some suggestions for further modifications to the design. The task force has carefully reviewed these suggestions, and the conclusions from this discussion are added.

9.6.1 Evaluator 1

9.6.1.1 Evaluation summary

The f8 Algorithm: The design of f8 appears sound. The best possible attack seems to be based on collisions and requires an amount of known plaintext blocks of the order 2^{32} or else has very low probability which decreases to roughly 2^{-52} in the context of 3GPP. The probability that information may leak appears negligible. Thus, the cryptanalytic scenario considered does not seem to endanger the security in the currently envisioned operational context of 3GPP. However, the evaluators wish to emphasise that their evaluation is strictly limited to this context. They do not recommend to extend the use of f8 to other applications.

The f9 Algorithm: The design of f9 appears sound. Using decorrelation theory, the security provided by f9 has been evaluated. Provided the underlying block cipher is close enough to random, there is a form of provable security based on the assumption that a given key is only used for authenticating an overall number of blocks which is (significantly) below 2^{32}. From a cryptanalytic perspective, the evaluators have considered a forgery attack that requires 2^{64} chosen messages. Even if better attacks can be found, it is quite unlikely that they might have dangerous consequences in the context of 3GPP.

The block cipher: The evaluators have reviewed the design principles on which the block cipher is built and they have checked its strength against linear and differential attacks. They have also investigated other types of cryptanalytic attacks, such as higher order attacks and impossible differentials. They have shown how distinct structural properties of the cipher could be combined in order to mount an “academic" attack against a version of the cipher reduced to six rounds. Finally, they have studied the key schedule and have found a related key attack on five rounds.

The evaluators have not found any attack on the full version of the cipher. On the other hand, they remark that the design of the cipher does not allow to provide a proof, e.g. along the lines of decorrelation theory. It might be the case that some other tricks exist, which have not been discovered in the present analysis, and which would allow to further “pile up” the imperfections of the cipher. Accordingly, even if none of their observations seems to endanger the security in the operational context of 3GPP, the evaluators have suggested that the designers consider minor modifications that should eliminate the imperfections that they have spotted.

9.6.1.2 Task force response

The main recommendation was for the (re)consideration of the addition of a fourth round to FO.

The task force accepts that such a change would improve the security margin of KASUMI and make it more resistant to certain categories of attack. However, such a change is a non-trivial exercise as, amongst other things, it would require a significant change to the key scheduling and would be likely to add to the overall complexity (hardware gate count). The task force revisited this at length and finally concluded that, whilst they could see a variety of benefits from such a change, they felt that the existing security margins were adequate *within the context of 3GPP* and they could see no real justification for delaying the delivery of the algorithms whilst this change was made.

9.6.2 Evaluator 2

9.6.2.1 Evaluation summary

The block cipher: KASUMI itself is based on the block cipher MISTY, which was designed by Matsui, a highly respected cryptographer, of the Mitsubishi Electric Corporation. Since MISTY has been widely studied without the detection of any weaknesses, the evaluators are confident that KASUMI has a sound 'foundation'. They have found nothing to suggest that the changes make KASUMI any less secure as a block cipher than MISTY1. However, they are not always sure why certain changes to MISTY have been made and, for some changes, do not have enough evidence that provides assurance that the changes do not induce any weaknesses.

The evaluators considered the application of the several well-known cryptanalytic attacks to KASUMI, but were unable to find any significant results. They utilised some of the design properties and some of the statistical data given in *Statistical Evaluation of 3GPP Confidentiality and Integrity Algorithms* in order to evaluate the effectiveness of some of the following attacks.

- Divide and Conquer Attacks
- Meet-in-the-Middle Attack
- Differential Cryptanalysis (and Variants)
- Linear Cryptanalysis
- Related Key Attacks
- Weak Key Classes

In each case they did not detect any exploitable properties and thus have nothing to report other than to confirm that they tried.

The f8 and f9 algorithms: The evaluators have some reservations about the use of key modifiers with KASUMI. In f8 the first keystream block is obtained by performing a double KASUMI encryption under highly similar subkeys. Also in f9 the related nature of the subkeys of IK and IK+KM gives cause for concern should KASUMI have a weakness in respect of such double encryptions. Whether this property leads to a weakness appears not to have been investigated or tested statistically. This raises concern and suggests that some appropriate tests should be designed and implemented.

Conclusions: The evaluators have been unable to detect any weaknesses in the proposed algorithms. However they have a number of reservations that they feel need addressing. Part of the reason for their reservations relates to the fact that they did not feel they had long enough to perform a serious assessment. The rest relates to certain features that appear to be 'unusual' and possibly exploitable. The evaluators have no positive reason for concern and are confident that, since they appear to be deliberately designed features, the designers must have studied them in detail. The evaluators would like to see justifications for the following:

- Key Adequacy
- A number of apparently minor changes have been made to MISTY, especially the key schedule. The evaluators are not sure what advantages these changes bring and would like to see evidence that they do not introduce weaknesses.
- Non-standard methods of MACing and chaining. The evaluators can see no problem with the chaining methods proposed but would like to see the designers' analysis of the proposed methods with KASUMI.

9.6.2.2 Task force response

The task force hopes that the detail in this Public Report will adequately answer the most of the points raised, but, taking the three main issues from the conclusion:

Key Adequacy

This part is outside the scope of the current design and evaluation work. Neither the *f3* or *f4* functions are being standardised; Network Operators are free to select their own algorithms for these functions. The designers have no control over the way in which Network Operators run their individual systems. As a design team their aim is to ensure that, if the key material is derived in a suitable manner, each algorithm delivers its claimed strength.

In retrospect maybe a valid assumption might have been to consider the case where CK=IK. (i.e. can *f8* be attacked in combination with *f9*?). Other weaknesses in the *f3* and *f4* functions (such as using the random challenge directly) remove the need for any strength in the *f8* and *f9* algorithms.

Changes to Misty

The design team has carefully considered the changes that they have made to Misty and their consequences. Many of the reasons are included in this Public Report. The major change, the simplification of the key schedule, was made to simplify the realisation of the algorithm in hardware so reducing the required gate count and power consumption. Some of the other changes (e.g. the fourth round in FI) have been introduced to counter minor weaknesses that have been found by the evaluation team.

Non-standard MACing and chaining

The modes of *f8* and *f9* have evolved from more standard methods to counter minor concerns that were raised during the design and evaluation work.

The designers note the concern over the “double encryption” by Kasumi, but feel that weaknesses in this area would have emerged during the work on related key attacks. However the designers have instigated a further series of tests where Kasumi is treated as 16-round block cipher, with the key schedule for each set of 8 rounds being derived from CK+KM and CK respectively.

Extra statistical testing

The evaluators also recommended carrying out some extra statistical tests on the key modifiers for the f8 function. The Task Force implemented and carried out these tests. The results of the test were satisfactory.

9.6.3 Evaluator 3

9.6.3.1 Evaluation summary

A security evaluation has been performed of the proposed 3GPP encryption and integrity algorithms with an effort of approximately 20 person-days.

No flaws or security weaknesses have been identified in the underlying KASUMI block cipher, and the statistical evaluation by the designers has been deemed satisfactory. The proposed key size of 128 bits offers sufficient protection against exhaustive search for the next 20 years or more. No practical attack has been found on a reduced version of KASUMI with five rounds, and only certificational weaknesses (that is, theoretical shortcut attacks) have been identified in a version reduced to six rounds. Therefore the evaluators believe that KASUMI with eight rounds offers a sufficient security margin against current cryptographic techniques. In view of the fast evolutions in cryptography, they recommend to update this evaluation at least every five years.

No weaknesses have been identified in the encryption and integrity modes (f8 and f9). For the application in which the block cipher KASUMI will be used, a sufficiently high security level is provided. It would be advisable to limit the number of initial values for a single key to about 50 million (this should not be a practical constraint, as this corresponds to 40 hours at a speed of 2 Mbit/s).

Some minor improvements are proposed to the mode in which the algorithm is being used (usage with short keys, and integrity algorithm).

The evaluators have not determined any properties of KASUMI that would make implementations of KASUMI particularly vulnerable to implementation attacks such as timing or power analysis.

The scope of the review was limited to cryptanalysis of the block cipher KASUMI as described in [5], and its immediate use as a component of the functions f8 and f9 for use within the access link as described in [4].

Finally the evaluators observe that their report is the result of only a limited time of review. Others with more time and dedicated resources may well find attacks that they have not been able to identify during this time.

9.6.3.2 Task force response

Constants

The evaluators express some concern regarding the regular structure of the constants used in the key schedule. The designers take the point raised here but feel that it does not justify any change to the algorithm on its own. Should any changes be made to the algorithm, then this point would be addressed. The main point in this area is that it should be clear that the constants have been selected in such a way that there is no hidden property being deliberately introduced into the algorithm.

Short Keys

The requirement specification [2] states that if the key should need to be shorter than 128 bits, the least significant bits of the key shall be set to zero. The evaluators find that it is conceivable that some advantage might be gained by an attacker in this situation.

The designers accept this point. A liaison statement will be sent to the relevant 3GPP committee recommending that the current wording in the requirement specification be replaced by: "Where the operational key length is less than 128 bits the operational key should be repeated as many times as is necessary to build up the 128-bit key required by the algorithms".

Length in MAC calculation

The evaluators propose to include the input length in the beginning of the message before calculating the MAC.

This was considered during the design phase and a decision was made not to include the length field for a number of operational reasons. In some applications it may be preferable to append the length rather than prepend it for practical reasons. The designers reconsidered the decision in the light of the observations made in the evaluation report.

The designers note that the MAC is being used to protect signalling messages, messages that are assumed to have a high degree of structure. The designers feel that the primary concern is bit modification (or replay). Any probability of an attack involving block insertion having a practical impact is further reduced by this inherent structure. For these reasons the designers have decided that, on its own, this concern does not justify a change to the algorithm.

9.7 Results from complexity evaluation

Independent manufacturers have tested the implementation complexity of the 3GPP confidentiality and integrity algorithms. Their finding concludes that the proposed algorithms fall within the requirements specified in section 6. A high-level realisation of KASUMI has been conducted. The conclusion is that the circuit size is manageable and below 3000 gates.

9.8 Conclusion of evaluation

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 carefully 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.

Annex A: External references

[9] 3GPP TS 25.321: "3rd Generation Partnership Project; Technical Specification Group (TSG) RAN; MAC protocol specification".

[10] S. Ar, R.J. Lipton, R. Rubinfeld, and M. Sudan: "Reconstructing algebraic functions from mixed data", *SIAM J. of Comput*., Vol. 28, No. 2, 1998, pp 487-510.

[11] E. Biham, A. Biryukov, and A. Shamir: "*Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials*", Technical Reports of the Computer Science Department in the Technion, 0947.

[12] Hans Dobbertin: "Almost Perfect nonlinear Power Functions on GF(2^n): The Welch case", *IEEE Transactions on Information Theory*, Vol. 45, NO.4, May 1999

[13] T. Jakobsen: "*High-Order Cryptanalysis of Block Ciphers*", PhDThesis, Department of Math., Technical University of Denmark, 1999.

[14] L. Knudsen: "*Truncated and Higher Order Differentials*", Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008, Springer Verlag, 1995, pp. 196-211.

[14a] L. Knudsen, C Mitchell: "*An analysis of the 3gpp-MAC scheme*", private communication, May 25, 2000.

[15] Donald E. Knuth: "*The Art of Computer Programming*", *Volume 2, Seminumerical Algorithms*, Addision-Wesley, Third Edition, 1998.

[16] A.G. Konheim: "Cryptography, a primer" , NY, John Wiley & Sons, 1981.

[17] X. Lai. "*Higher order derivatives and differential cryptanalysis*", In Proc. "Symposium on Communication, Coding and Cryptography" in honour of James L. Masseyon the occasion of his 60'th birthday, Feb. 10-13, 1994, Monte – Verita, Ascona, Switzerland, 1994.

[18] S. K. Langford and M. E. Hellman, "*Differential – Linear Cryptanalysis*", Advances in Cryptology – CRYPTO '94, in Lecture Notes in Computer Science 839, Springer, pp. 17-25.

[19] Mitsuru Matsui: "*New Block Encryption Algorithm MISTY*", Proceedings of Fast Software Encryption '97 conference, in Lecture Notes in Computer Science 1267, Springer, pp. 54-68.

[20] Willi Meier, Othmar Staffelbach, "*Nonlinearity Criteria for Cryptographic Functions*", EUROCRYPT' 89

[21] A. Menezes, P.C. van Oorschot, S.A Vanstone, "*Handbook of Applied Cryptography*", CRC Press, 1997.

[22] Kaisa Nyberg, "*Differential uniform mappings for cryptography*", EUROCRYPT' 93.

[23] K.Nyberg and L. Knudsen, "*Provable Security Against a Differential Attack*", Journal of Cryptology Vol 8 Nr 1, 1995

[24] S. Luck, "*On the Security of the 128-Bit Block Cipher DEAL*", http://th.informatik.uni-mannheim.de/m/lucks/papers/deal.ps.gz

[25] Rainer A. Rueppel, "*Analysis and Design of Stream Ciphers*", Springer, 1986.

[26] SSLeay source, ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL (1999), see also http://www.cryptsoft.com/ssleay/faq.html (1999).

[27] M. Sudan, "*Decoding of Reed-Solomon codes beyond the error-correction bound*", Journal of Complexity, Vol. 13, 1997, pp 180-193.

[28] M. Sugita: “*Higher Order Differential Attack on Block Cipher MISTY1*, 2”. Technical Report of IEICE, ISEC98-4, May 1998.

[29] Hidema Tanaka, Kazuyuki Hisamatsu and Toshinobu Kaneko*: “Strength of MISTY1 without FL function for Higher Order Differential attack*”. The 3rd World Multiconference on Systemic Cybernetics and Informatics and the International Conference on Information Systems Analysis and Synthesis SCI'99/ISAS'99, Orlando, USA, August 1999.

[30] TSG SA WG3#5: Liaison Statement to SA3 on Ciphering Algorithm Requirements by RAN WG2.

[31] D. Wagner, “*The Boomerang Attack”*, FSE '99, to appear.

[32] Wassenaar Arrangement, December 1998.

Annex B: Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
10-2000					Finalisation of evaluation report results by ETSI SAGE. Edited into 3GPP TR format for submission to SA WG3 / SA#10 as 33.909	SAGE v2.0	0.0.1