|  |
| --- |
| 3GPP TS 35.249 V19.1.0 (2025-07) |
| Technical Specification |
| 3rd Generation Partnership Project;Technical Specification Group Services and Security Aspects;Specification of an example algorithm for f5\*\* function for MILENAGE and TUAK;Algorithm specification and test data(Release 19) |
|   |
|  |  |
|  |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. |

|  |
| --- |
|  |
| ***3GPP***Postal address3GPP support office address650 Route des Lucioles - Sophia AntipolisValbonne - FRANCETel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16Internethttp://www.3gpp.org |
| ***Copyright Notification***No part may be reproduced except as authorized by written permission.The copyright and the foregoing restriction extend to reproduction in all media.© 2025, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).All rights reserved.UMTS™ is a Trade Mark of ETSI registered for the benefit of its members3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersLTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersGSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 7

3.1 Terms 7

3.2 Symbols 7

3.3 Abbreviations 7

4 Structure of this specification 7

5 Introductory information 8

5.1 Notation 8

5.1.1 Radix 8

5.1.2 Bit ordering and arrays 8

6 List of variables 8

6.1 General 8

6.2 Input/output variables specific for f5\*\* 9

6.3 Input/output parameter sizes for f5\*\* 10

7 General definition of f5\*\* 10

8 MILENAGE-128 based f5\*\* 10

8.1 Supported parameter sizes 10

8.2 Algorithm specification 10

8.2.1 General 10

8.2.2 f5\*\* specification 10

9 Tuak based f5\*\* 11

9.1 Supported parameter sizes 11

9.2 Algorithm specification 11

9.2.1 General 11

9.2.2 f5\*\* specification 11

10 Security analysis 12

10.1 Resynchronisation attack description and impact 12

10.2 Analysis 12

10.2.1 General 12

10.2.2 MILENAGE-128 based f5\*\* 13

10.2.3 Tuak based f5\*\* 13

11 Implementors’ test data 14

11.1 MILENAGE-128 variant 14

11.2 Tuak variant 15

Annex A (informative): Change history 19

# Foreword

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

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document presents an optional security enhancement for the MILENAGE [5, 6, 12, 13] and Tuak example algorithm set [7, 14, 15] for the 3GPP authentication and key agreement functions. The enhancement addresses a subscriber traceability issue discovered by academic researchers [9] and is provided as an alternative anonymity key generation function f5\*\* that, when enabled, replaces the f5\* function of the aforementioned example algorithm sets. A functionally equivalent security enhancement is already included in the new MILENAGE-256 example algorithm set [8].

# 2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 33.102: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security Architecture".

[3] The Advanced Encryption Standard (AES), NIST FIPS 197, NIST, 2001.

[4] Rijndael information page, NIST archived AES submissions, https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development#rijndael.

[5] 3GPP TS 35.205: "Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 1: General".

[6] 3GPP TS 35.206: "Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 2: Algorithm Specification".

[7] 3GPP TS 35.231: "Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 1: Algorithm specification".

[8] ETSI SAGE, "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP Authentication and Key Generation functions f1, f1\*, f2, f3, f4, f5, f5\* and f5\*\*, Document 1: General".

[9] R. Borgaonkar, "New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols", in Proceedings on Privacy Enhancing Technologies 2019(3):108-127. Also available at https://eprint.iacr.org/2018/1175.pdf (published online: July 2019).

[10] ETSI TR 133 909, "Report on the design and evaluation of the MILENAGE algorithm set; Deliverable 5: An example algorithm for the 3GPP authentication and key generation functions", 2001.

[11] ETSI SAGE: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP Authentication and Key Generation functions f1, f1\*, f2, f3, f4, f5, f5\* and f5\*\*, Document 4: Summary and Results of Design and Evaluation".

[12] 3GPP TS 35.207: "Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 3: Implementors' test data".

[13] 3GPP TS 35.208: "Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 4: Design conformance test data".

[14] 3GPP TS 35.232: "Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 2: Implementers’ test data".

[15] 3GPP TS 35.233: "Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 3: Design conformance test data".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

## 3.2 Symbols

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

 Å The bitwise exclusive-OR operation

X[i] The ith bit of the variable X. (X = X[0] || X[1] || X[2] || ….. ).

|| Concatenation: X[0]X[1]…X[n-1] || Y[0]Y[1]…Y[m-1] = X[0]X[1]…X[n-1]Y[0]Y[1] …Y[m-1]

rot(x,r) The result of cyclically rotating the 128-bit value **x** by **r** bit positions towards the most significant bit. See [6].

MILENAGE-128 The MILENAGE algorithm set as defined in [5, 6]. NOTE: The term MILENAGE-128 is used to distinguish this algorithm set from the more recently defined MILENAGE-256 set [8].

E[x]k The AES-based cryptographic kernel of MILENAGE-128. Specifically, the result of applying the block cipher E (AES) to the input value **x** using the key **k**.

Tuak The Tuak algorithm set as defined in [7].

П The Keccak-based cryptographic kernel of Tuak.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AES Advanced Encryption Standard

AKA Authentication and Key Agreement

MAC Message Authentication Code

ETSI SAGE ETSI Security Algorithms Group of Experts

# 4 Structure of this specification

This specification is organised as follows:

Clause 3 and 5 introduce symbols and notation used in the subsequent clauses.

Clause 6 provides a summary of all variables (inputs, outputs, and intermediary values) used in the algorithm specification.

Clause 7 provides a general definition of the f5\*\* function which is in agreement with the API of [2, Annex J] and which is independent on the underlying cryptographic kernel, i.e. it applies to both the instantiation within the MILENAGE-128 framework as well as with Tuak.

Clause 8 and 9 contains the specifics of the MILENAGE-128 and Tuak instantiations, respectively.

Clause 10 provides a security analysis.

Clauses 11 contains test data for implementors of the MILENAGE-128 and Tuak-based variants.

NOTE: The reader is reminded of the fact that both MILENAGE-128, as well as Tuak, in principle allow distinct choices for the security primitive (block cipher or hash function, respectively) employed as cryptographic kernel, provided the same input/output parameters can be securely supported. However, the present document assumes that the default kernels are used, i.e. that AES/Rijndael [3, 4] is used to instantiate MILENAGE-128 [5, 6] and the Keccak-based kernel of [7] is used to instantiate Tuak.

# 5 Introductory information

## 5.1 Notation

### 5.1.1 Radix

Unless otherwise noted, integer values are represented in decimal. The prefix **0x** indicates **hexadecimal** integers.

### 5.1.2 Bit ordering and arrays

The same ordering as in [5, 6, 7] shall apply. In other words, all data variables in the present document are presented with the most significant substring on the left hand side and the least significant substring on the right hand side. A substring may be a bit, byte or other arbitrary length bit string. Where a variable is broken down into a number of substrings, the left-most (most significant) substring is numbered 0, the next most significant is numbered 1, and so on through to the least significant.

EXAMPLE: **RAND**[0] is the most-significant bit of **RAND** and **RAND**[127] is the least significant bit of **RAND**.

# 6 List of variables

## 6.1 General

The set of variables defined in [5, 6, 7] shall apply also for the purposes of the present document. The following variables are highlighted since they are of particular relevance for the definition of f5\*\*.

Table 6.1-1: List of variables

|  |  |
| --- | --- |
| Name | Comment |
| **AK** | An anonymity key that is output by the functions f5 and f5\* of [5, 6]. NOTE: The MILENAGE-256 algorithm set [8] distinguishes the **AK** output by f5 from those output by f5\* by using the variable **AK**\* for the latter function. This distinction is not made in [5, 6, 7] and is therefore not made in the present document either,to avoid confusion when cross-referencing these documents. |
| **AMF** | A two byte of authentication management field that is input to the functions f1 and f1\*. |
| **K** | A subscriber key that is an input to the functions f1, f1\*, f2, f3, f4, f5, f5, f5\*. |
| **MAC-S** | A resynchronisation authentication code that is output by the function f1\*. |
| **RAND** | A random challenge that is an input to the functions f1, f1\*, f2, f3, f4, f5, f5\*. |
| **SQN** | A sequence number that is an input to either of the functions f1 and f1\*. (For f1\* this input is more precisely called **SQN**MS) |
| **c1, …, c5** and **r1, .., r5** | Operator selectable, 128-bit algorithm customization constants used in thedefinition of MILENAGE-128. These are not used in Tuak. |
| **TEMP** | An intermediate 128-bit value occurring during MILENAGE-128 computation. This value is not used by Tuak. |
| **IN** | Input value formed during the computation of the f-functions. **IN** is 128 bitsin MILENAGE-128 and 1600 bits in Tuak. NOTE: Two specific **IN**-values of importance for the MILENAGE-128 variant of f5\*\* are below denoted **IN1** and **IN6**. |
| **OUT** | Output value obtained by invoking the cryptographic kernels of MILENAGE-128 or Tuak on the **IN** values, and from which the f-function outputs are obtained by selecting some or all bits. |
| **INSTANCE** | An 8-bit value used as part of the input values (**IN**) to Tuak and which isassigned different values for the different f-functions. This value is not usedby MILENAGE-128. |
| **OPc** | An algorithm customization value (derived from an operator selected valueOP). |

## 6.2 Input/output variables specific for f5\*\*

Table 6.2-1: Input/output variables specific for f5\*\*

|  |  |
| --- | --- |
| Name | Comment |
| **AK** | This variable name is reused to denote the anonymity resynchronisation keythat is output by the function f5\*\*. NOTE: Under normal operation, the **AK** value is provided by f5. If a resynchronisation procedure is required, an additional **AK** value is provided either by f5\*, or by f5\*\*. |
| **K** | The subscriber key is an input also to f5\*\*. |
| **MAC-S** | The resynchronisation authentication code that is output by the function f1\* and which is used also as an input to f5\*\*. |
| **RAND** | The random challenge, also used as an input to f5\*\*. |

## 6.3 Input/output parameter sizes for f5\*\*

Table 6.3-1: Input/output parameter sizes for to f5\*\*

|  |  |  |
| --- | --- | --- |
| Name | Permitted values | Comment |
| AKSZ | 48 | The size in bits of the anonymity key **AK**. |
| KSZ | Kernel-dependent | The size in bits of the subscriber key **K**. |
| MACSZ | Kernel-dependent | The size in bits of the resynchronisation authentication code **MAC-S**. |
| RANDSZ | 128 | The size in bits of the random challenge **RAND**. |
| SQNSZ | 48 | The size in bits of the sequence number **SQN**. |

# 7 General definition of f5\*\*

Annex J of [2] specifies an alternative f5\* function offering protection against the resynchronization attack of [9]. Its API is specified as follows:

Inputs: the subscriber key **K**, the network challenge **RAND**, and the resynchronisation MAC-value **MAC-S**.

Output: an anonymity key, **AK**.

In the present document, this function is defined as:

**AK** = f5\*\*(**K**, **RAND**, **MAC-S**).

NOTE: f5\*\* can also depend on other values, e.g. operator selected customization values, not shown.

# 8 MILENAGE-128 based f5\*\*

## 8.1 Supported parameter sizes

The MILENAGE-128 version of f5\*\* shall support KSZ = 128 bits and MACSZ = 64 bits only.

## 8.2 Algorithm specification

### 8.2.1 General

The choice to implement f5\*\* instead of f5\* has no impact on the f-functions (f1, f1\*, f2, f3, f4, and f5). The specifications [5, 6] shall apply except as specified in clause 8.2.2 below.

### 8.2.2 f5\*\* specification

Precisely the following modifications to clause 4.1 of [6] shall apply:

1. References to the function f5\* shall be ignored.

2. An additional 128-bit value **IN6** shall be computed as follows:

**IN6**[0] **IN6**[1]…**IN6**[63] = (**MAC-S**[0] Å 1) (**MAC-S**[1] Å 1) **MAC-S**[2]… **MAC-S**[63]

and

**IN6**[64] **IN6**[65]…**IN6**[127] = **MAC-S**[0] **MAC-S**[1]…**MAC-S**[62] **MAC-S**[63].

3. An additional value **OUT6** shall be computed as follows:

**OUT6** = E[**TEMP** Å rot(**IN6** Å **OPC**, **r1**) Å **c1**]K Å **OPC**.

4. Output of f5\*\* = **AK**, where **AK**[0] .. **AK**[47] = **OUT6**[0] .. **OUT6**[47]

NOTE 1: By modification 2, the two 64-bit halves of **IN6** are identical, except that the two bits **IN6**[0], **IN6**[1] are both inverted, relative to **IN6**[64], **IN6**[65].

NOTE 2: Conceptually, modification 3 defines two new constants **c6** and **r6**, but with values identical to **c1** and **r1**.

# 9 Tuak based f5\*\*

## 9.1 Supported parameter sizes

The Tuak version of f5\*\* shall support KSZ = 128 and 256 bits and shall support MACSZ = 64, 128, and 256 bits. The chosen values of these quantities shall be the same as those commonly chosen for the complete algorithm set.

## 9.2 Algorithm specification

### 9.2.1 General

The choice to implement f5\*\* instead of f5\* has no impact on the f-functions f1, f1\*, f2, f3, f4, and f5. The specification [7] shall apply except as specified in clause 7.2.2 below.

### 9.2.2 f5\*\* specification

Clause 6.5 of [7] shall be replaced by the following. Recall that the bit-length of **MAC-S** is defined as MACSZ.

The internal **INSTANCE** variable shall be constructed as follows:

**INSTANCE**[0] ... **INSTANCE**[1] = 1,1

**INSTANCE**[2] ... **INSTANCE**[4] = 0,0,1 if MACSZ is 64 bits,

= 0,1,0 if MACSZ is 128 bits,

= 1,0,0 if MACSZ is 256 bits

**INSTANCE**[5] ... **INSTANCE**[6] = 0,0

**INSTANCE**[7] = 0 if the length of **K** is 128 bits

= 1 if the length of **K** is 256 bits.

Given **INSTANCE** as above, this value shall be assigned to **IN**[256] .. **IN**[263]. The rest of the 1600-bit value **IN** (i.e. **IN**[0]..**IN**[255] and **IN**[264]..**IN**[1599]) shall then be constructed in exactly the same way as for f2, f3, f4, and f5 except that

**IN**[768] … **IN**[768+ MACSZ – 1]=**MAC-S**[MACSZ – 1] ... **MAC-S**[0],

**IN**[i] = 0, for 768 + MACSZ ≤ i ≤ 1023,

**IN**[i] = 1 for 1024 ≤ i ≤ 1028.

Then the permutation:

**OUT** = Π(**IN**)

is applied and the function output is extracted as follows:

Output of f5\*\* = **AK**, where:

**AK**[0] .. **AK**[47] = **OUT**[815] .. **OUT**[768].

NOTE 1: The value **INSTANCE**[0] **INSTANCE**[1] is identical to that defined for f5\* but **INSTANCE**[2] .. **INSTANCE**[4] is different.

NOTE 2: For f2 to f5, the input padding starts at bit **IN**[768], while for f5\*\*, the padding starts after **MAC-S** has been zero-padded to 256 bits, i.e. in bit **IN**[1024].

# 10 Security analysis

## 10.1 Resynchronisation attack description and impact

The attack presented in [8] can be summarised as follows. An attacker, Eve, has previously recorded an authentication between a victim, Bob, and the network. This resulted in a resynchronisation procedure. (In fact, Eve can force such a resynchronisation by replaying a previously observed authentication challenge, known to be directed to Bob.) Thus, Eve has at time T1 obtained a value of the form

**SQNBob** (T1) Å f5\*(**KBob**, **RAND**(T1))

as well as the network authentication token **AUTN**(T1), used at that time.

Later, at time T2, Eve wants to test if a certain subscriber is Bob or someone else, so she replays (**RAND**(T1), **AUTN**(T1)).

If it is not Bob, the UE/USIM will generate a "network authentication failure", observable to Eve and she can exclude that it is Bob.

On the other hand, if it is Bob, Bob's UE/USIM will initiate a resynchronisation (also observable to Eve) and this will reveal **SQNBob**(T2) Å f5\*(**KBob**, **RAND**(T1)). Taking the XOR of these values, Eve obtains

**SQNBob**(T1) Å **SQNBob**(T2).

Assuming Eve knows the sequence number generation method, at least some bits of this value will have a low Hamming weight since the values are likely related. In fact, if Bob has not performed any authentication since time T1, the XOR-value will be identically zero.

In conclusion, this provides Eve with a tracing mechanism for certain targeted subscribers.

## 10.2 Analysis

### 10.2.1 General

To mitigate the attack, it is possible to use a more sophisticated concealment mechanism for the **SQN** value than a simple XOR. However, this requires quite substantial modifications.

Another possibility is to add some freshness to the input of f5\*, making it likely that the outputs are different. Without major changes to the AKA protocol, the only source of such freshness is the **MAC-S** value, which has led to an alternative f5\* function of the form

**AK** = f5\*\*(**K**, **RAND**, **MAC-S**)

akin to the two variants defined in the present document. There still remain two "bad" events that would allow Eve to trace Bob:

1. The **MAC-S** values are derived by f1\*, with dependence on **RAND** and **SQN**, and could still collide "by chance" even if the (**RAN**, **SQN**)-values are distinct. This however has a low probability, exponentially small in the bit-size of **MAC-S**.

2. As noted above, if the resynchronisation occurs without Bob having performed any successful authentication in the time-interval (T1, T2), **SQN** will still have the same value, causing **AK**-values to collide, which Eve can detect. There seems however to be no way to avoid this without substantial changes to the AKA protocol, and therefore the f5\*\* approach above has been deemed a good compromise with sufficient protection and low additional complexity.

NOTE: This approach to the resynchronisation protection is conceptually very similar to known solutions that protect stream-ciphers against nonce-reuse and the construction is therefore judged to be cryptographically sound. More discussion on this can be found in [11].

### 10.2.2 MILENAGE-128 based f5\*\*

The main issue is the analysis of the possibility that f5\*\* output could collide with one of the other f-functions. This is in turn dependent on the distinctness of the f5\*\* input:

TEMP Å rot(**IN6** Å OPC, r1) Å c1 (1)

from values of form

TEMP Å rot(**IN1** Å OPC, r1) Å c1, (2)

i.e. the input to **f1/f1**\*, as well the distinctness between (1) and values of form

rot(TEMP, ri) Å ci, i = 2, 3, 4, 5, (3)

i.e. the inputs occurring when computing f2-f5. This is analysed in great detail in [10], an analysis which largely applies also here. Let 3128 be the 128-bit representation of the integer value 3 \* 2126 i.e. the binary string 1100…0.

• The distinctness of (1) relative to the values (3) is ensured by parity requirements:

o Rotation does not affect parity.

o As required by [6], c2, c3, c4, c5 all have odd parity, while c1 has even parity.

o The value **IN6** has structure (Y || Y) Å 3128 (where Y = **MAC-S**). The part (Y || Y) always has even parity and this is preserved by the exclusive-or with 3128.

o Together, this implies that regardless of the parity of TEMP, the values involved in (1) and (3) can never have equal parity and are therefore distinct.

• The distinctness from the value TEMP Å rot(**IN1** Å OPC, r1) Å c1 of (2) follows since **IN1** by [6] has format AMF || SQN || AMF || SQN, i.e. is a value of format X || X. But such a value can never be identical to a value of format (Y || Y) Å 3128, such as **IN6**. Since the same (c1, r1)-value is used both for f5\*\* and f1/f1\*, the distinctness follows.

In conclusion, collision probabilities are not affected by f5\*\*.

### 10.2.3 Tuak based f5\*\*

Also here, the only additional concern is the possibility that the input to f5\*\* could collide with the input when computing one of the other f-functions. The absence of such collisions is ensured by the **INSTANCE**-values included in the inputs, which are always distinct.

While f5\*\* uses the same value for **INSTANCE**[0]**INSTANCE**[1] as f5\* does, the values of **INSTANCE**[2]..**INSTANCE**[4] are never identical for f5\* and f5\*\* (they are always identically zero for f5\*).

NOTE: Furthermore, only one of f5\* and f5\*\* is assumed to be supported.

# 11 Implementors’ test data

## 11.1 MILENAGE-128 variant

TEST SET #1

K 465b5ce8 b199b49f aa5f0a2e e238a6bc

RAND 23553cbe 9637a89d 218ae64d ae47bf35

SQN ff9bb4d0 b607

AMF b9b9

OP cdc202d5 123e20f6 2b6d676a c72cb318

OPc cd63cb71 954a9f4e 48a5994e 37a02baf

f1 4a9ffac3 54dfafb3

f1\* 01cfaf9e c4e871e9

f2 a54211d5 e3ba50bf

f5 aa689c64 8370

f3 b40ba9a3 c58b2a05 bbf0d987 b21bf8cb

f4 f769bcd7 51044604 12767271 1c6d3441

f5\* 451e8bec a43b

f5\*\* 4edd7fbd c382

TEST SET #2

K 0396eb31 7b6d1c36 f19c1c84 cd6ffd16

RAND c00d6031 03dcee52 c4478119 494202e8

SQN fd8eef40 df7d

AMF af17

OP ff53bade 17df5d4e 793073ce 9d7579fa

OPc 53c15671 c60a4b73 1c55b4a4 41c0bde2

f1 5df5b318 07e258b0

f1\* a8c016e5 1ef4a343

f2 d3a628ed 988620f0

f5 c4778399 5f72

f3 58c433ff 7a7082ac d424220f 2b67c556

f4 21a8c1f9 29702adb 3e738488 b9f5c5da

f5\* 30f11970 61c1

f5\*\* 7b958d44 d816

TEST SET #3

K fec86ba6 eb707ed0 8905757b 1bb44b8f

RAND 9f7c8d02 1accf4db 213ccff0 c7f71a6a

SQN 9d027759 5ffc

AMF 725c

OP dbc59adc b6f9a0ef 735477b7 fadf8374

OPc 1006020f 0a478bf6 b699f15c 062e42b3

f1 9cabc3e9 9baf7281

f1\* 95814ba2 b3044324

f2 8011c48c 0c214ed2

f5 33484dc2 136b

f3 5dbdbb29 54e8f3cd e665b046 179a5098

f4 59a92d3b 476a0443 487055cf 88b2307b

f5\* deacdd84 8cc6

f5\*\* e7fd8260 d2c9

TEST SET #4

K 9e5944ae a94b8116 5c82fbf9 f32db751

RAND ce83dbc5 4ac0274a 157c17f8 0d017bd6

SQN 0b604a81 eca8

AMF 9e09

OP 223014c5 806694c0 07ca1eee f57f004f

OPc a64a507a e1a2a98b b88eb421 0135dc87

f1 74a58220 cba84c49

f1\* ac2cc74a 96871837

f2 f365cd68 3cd92e96

f5 f0b9c08a d02e

f3 e203edb3 971574f5 a94b0d61 b816345d

f4 0c4524ad eac041c4 dd830d20 854fc46b

f5\* 6085a86c 6f63

f5\*\* f0e42c7f af58

TEST SET #5

K 4ab1deb0 5ca6ceb0 51fc98e7 7d026a84

RAND 74b0cd60 31a1c833 9b2b6ce2 b8c4a186

SQN e880a1b5 80b6

AMF 9f07

OP 2d16c5cd 1fdf6b22 383584e3 bef2a8d8

OPc dcf07cbd 51855290 b92a07a9 891e523e

f1 49e785dd 12626ef2

f1\* 9e857903 36bb3fa2

f2 5860fc1b ce351e7e

f5 31e11a60 9118

f3 7657766b 373d1c21 38f307e3 de9242f9

f4 1c42e960 d89b8fa9 9f2744e0 708ccb53

f5\* fe2555e5 4aa9

f5\*\* 7468dacf 4f72

TEST SET #6

K 6c38a116 ac280c45 4f59332e e35c8c4f

RAND ee6466bc 96202c5a 557abbef f8babf63

SQN 414b9822 2181

AMF 4464

OP 1ba00a1a 7c6700ac 8c3ff3e9 6ad08725

OPc 3803ef53 63b947c6 aaa225e5 8fae3934

f1 078adfb4 88241a57

f1\* 80246b8d 0186bcf1

f2 16c8233f 05a0ac28

f5 45b0f69a b06c

f3 3f8c7587 fe8e4b23 3af676ae de30ba3b

f4 a7466cc1 e6b2a133 7d49d3b6 6e95d7b4

f5\* 1f53cd2b 1113

f5\*\* 74f55f33 b347

## 11.2 Tuak variant

Inputs to these test data are identical to those 6 test sets given in Clauses 6.3 to 6.8 of TS 35.232 [14]. Here only the delta-additions related to f5\*\* function are listed.

====================================================

6.3 Test set 1 [3GPP TS 35.232 V17.0.0 (2022-03)]

====================================================

+Intermediate Values:

IN when computing f5\*\*:

ff cb cc 40 1f 5d b4 3e a7 05 53 11 0c 33 e2 a8 23 46 95 ad c2 7a 83 5d 3c 51 87 0e

53 d9 04 bd c8 30 2e 31 4b 41 55 54 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42 42

00 00 00 00 00 00 00 00 ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab ab 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 f3 7d 29 c7 c6 4d 4b e9 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1f 00 00 00 00 00 00 80 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00

OUT when computing f5\*\*:

9e 37 8b 43 2b 04 6f 39 40 f4 d9 e4 32 27 0d da 10 bc 5c 9f d6 c7 fb f7 5a 44 7f e2

16 a2 23 e6 d4 08 3f 37 84 e7 1f 28 84 77 7c d8 94 b7 71 9d 5d b8 db 1e 19 a0 20 2d

2f a4 ab 1b 24 8c 89 9e fe 23 d2 c8 31 e5 1e c4 80 58 8a ba 7c b8 80 c7 ce 2f 15 20

8a 91 26 be 7e 5f a5 d1 8e 5d d1 82 4f 66 18 a4 62 7d 0c 1f 72 37 8d 45 35 d7 26 25

13 6f ef 93 0a a5 45 f9 71 ff 5a e3 08 98 d5 a6 bf a2 97 e9 a1 b9 92 a6 9d d8 ad 80

e0 2b d0 1a 00 9a 31 79 8e eb 4f fd 94 15 00 65 99 86 d8 ae 44 6c d0 e4 7c 9d 45 ac

30 4a f2 5e e5 49 61 f2 3e 1d 18 59 60 b4 b2 eb dc 77 0f 1a 35 38 be e7 53 ec c5 7b

2b 06 73 f8

+Output Parameters:

f5\*\*: 7d62a418664f

====================================================

6.4 Test set 2 [3GPP TS 35.232 V17.0.0 (2022-03)]

====================================================

+Intermediate Values:

IN when computing f5\*\*:

24 85 7f 97 d5 74 59 de 65 0c 9d b2 c6 c6 36 0c 5d c9 72 ea 94 b2 a4 c8 03 c5 18 7e

42 25 54 30 d1 30 2e 31 4b 41 55 54 ef cd ab 89 67 45 23 01 ef cd ab 89 67 45 23 01

00 00 00 00 00 00 00 00 e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef f0 f1 f2 f3

f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff 5b 74 a0 7f 53 fa ea 6c 2c 84 f7 90 72 af 81 ef

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1f 00 00 00 00 00 00 80 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00

OUT when computing f5\*\*:

c6 58 e8 84 cb bf 0d 05 b8 26 ae a7 ff e6 5b 48 b9 f2 70 b1 39 ae 2b d0 79 94 93 bc

a4 9f 54 95 23 b8 87 4c 6a 72 d1 87 ae 44 0d 16 6a f8 31 ed 7c 81 f7 4a 00 03 1e fd

4a 86 a7 90 a0 ba 05 10 90 af 78 a1 2d 25 97 3b 77 00 44 17 8c 6e 52 d2 16 fd 71 08

6e 56 98 fe 62 7b ce 11 98 eb 05 94 83 07 69 7a 46 be 7a 25 52 c9 00 2c 6c 62 9d ab

81 da 76 4f 7f dd 16 c4 c0 8e fb 09 d8 bd 12 77 a5 b6 68 4b 95 92 4c e6 7f 88 38 bf

a4 91 a6 e0 ef 6a 92 eb c0 63 74 96 24 d3 ae 9c 91 06 d6 b8 33 76 3e 4a 16 0c 9f 28

1b 98 63 ce c3 97 e9 9a ed 40 49 77 cc 05 c2 4a 18 b4 0e d3 52 d5 92 97 07 b7 f0 89

66 33 04 97

+Output Parameters:

f5\*\*: be467a690783

====================================================

6.5 Test set 3 [3GPP TS 35.232 V17.0.0 (2022-03)]

====================================================

+Intermediate Values:

IN when computing f5\*\*:

24 85 7f 97 d5 74 59 de 65 0c 9d b2 c6 c6 36 0c 5d c9 72 ea 94 b2 a4 c8 03 c5 18 7e

42 25 54 30 e1 30 2e 31 4b 41 55 54 ef cd ab 89 67 45 23 01 ef cd ab 89 67 45 23 01

00 00 00 00 00 00 00 00 e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef f0 f1 f2 f3

f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff 21 b8 11 81 25 50 15 5b 23 fe c9 64 a1 d4 6f 9a

90 f3 99 64 21 c5 f8 54 6c a8 e3 c6 07 bf 7b 42 1f 00 00 00 00 00 00 80 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00

OUT when computing f5\*\*:

d9 17 a7 73 1d d7 80 7c db 9e 36 ba b1 70 34 a3 b6 1c 4f 85 e1 c2 ac 37 dd bb f4 43

e4 1a 69 a7 f4 a7 e1 a3 22 20 90 20 ef 80 d0 c4 06 34 40 f3 11 b9 29 41 fd 03 72 12

da 5d be d0 31 1a f8 c6 1c 83 f4 28 c1 0f ec 69 fd 46 d9 54 a9 64 6a d9 7b 35 fe 82

2c 35 e7 53 44 d4 a9 44 49 63 70 08 df 6d ef 85 5c de 59 ee 2c ea d8 e2 23 ce f6 63

2c 86 60 59 d4 35 7a 65 dc b3 ba 5e ef 3b 7f ba 58 46 0b ef 07 b1 fd 1f 86 dc d4 9c

57 b7 57 97 24 49 aa d5 15 e7 5d d6 56 72 f4 d3 53 ad 3c 7a 13 99 d1 68 9e f0 57 59

1a a8 03 0b cf 14 06 06 6c d0 38 81 aa f6 50 3b 7d 6c d8 8e e9 8d f5 94 15 0a 63 85

af 0b 18 28

+Output Parameters:

f5\*\*: de5c85ef6ddf

====================================================

6.6 Test set 4 [3GPP TS 35.232 V17.0.0 (2022-03)]

====================================================

IN when computing f5\*\*:

0a 76 87 44 0e ca b6 1e 28 ce 52 e6 a2 27 35 49 1d fb 0e 7c f5 08 6f 44 1f 8e a6 57

b6 6e c1 2b d0 30 2e 31 4b 41 55 54 e8 2b a7 5f 1a 66 c9 86 bd 66 a9 25 54 e5 87 68

00 00 00 00 00 00 00 00 61 cc 6a 4f a1 7d c9 c7 6a 2d 65 50 7a 83 da b8 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 6d b5 df f9 63 30 e1 ae 82 e3 80 fe 5a 86 9e 61

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1f 00 00 00 00 00 00 80 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00

OUT when computing f5\*\*:

93 98 21 4b cc 29 23 f1 75 2b 64 bc 03 b9 e6 14 a3 a9 77 ba 35 f4 a1 13 e7 68 ce a8

01 f3 60 70 e9 0d 4f a3 f6 74 5e 07 7f 0a d2 02 db 67 f6 25 b0 a4 8f f2 2f f6 68 3d

cf f3 dc 79 4a 64 88 62 79 12 d4 ef f1 fe 75 7d ac 07 82 dd 8c 6c ac 00 a9 91 39 f2

69 5e 58 aa 80 86 bf 89 d9 37 ff 24 66 57 e4 11 ba b8 98 82 f5 5c b8 6a 98 c7 4c 59

cc 55 8e 18 fe 6f 08 19 5f e8 06 1f 1a 26 a5 28 9a 47 cb 22 90 ef 5d a2 33 fd 44 79

3c c4 a8 53 87 32 97 75 1d 4e 3d ea 47 4a 67 0c f4 da fd 56 a1 d7 ca 26 25 ad 41 67

57 f7 95 4c 7d e1 96 ce ef e6 8e 25 b9 5f e0 23 92 88 e4 2b 8e 90 9d 78 17 eb 92 4c

af 6c b7 9e

+Output Parameters:

f5\*\*: b8ba11e45766

====================================================

6.7 Test set 5 [3GPP TS 35.232 V17.0.0 (2022-03)]

====================================================

+Intermediate Values:

IN when computing f5\*\*:

62 ff 5e a5 65 61 7d 3e bc 48 cd ae 76 a9 aa f3 e8 23 f2 89 bb 3c aa 47 8a a2 32 15

e4 52 60 3c c9 30 2e 31 4b 41 55 54 ef 8b 49 22 83 08 e3 b1 1f 65 de 8c c6 aa 70 c5

00 00 00 00 00 00 00 00 a5 5a 0e 77 15 9f c2 69 2b aa 26 44 5f 95 44 42 cd c9 89 f7

80 28 c8 89 c1 05 1d 88 56 ca 74 15 15 cb ac 66 2e 1e 02 c6 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1f 00 00 00 00 00 00 80 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00

OUT when computing f5\*\*:

2b e1 c9 2a b7 9f 2b cc cb d9 32 97 c7 eb 3c 44 25 20 3f 86 31 b0 04 af 59 74 e7 70

a3 04 b4 bb 74 9f 00 57 42 d6 3e 78 7b ac 48 5f 8e 74 ce fc 99 85 cf f8 ef 2f 06 4e

28 ca 4b 4e 91 74 16 86 ec 0b 81 83 dd 9b 0c 87 8a 9c a0 df a6 b3 bb fb 06 14 b7 92

34 17 d4 10 4f 3e f5 26 76 5d 4d b9 37 a8 eb 86 8e 24 e4 fd d2 dd 4c 81 a4 66 47 b4

b2 e0 7c 5b 35 2b 86 5a 1d 43 73 54 42 77 85 3c b7 15 88 b2 fc b1 7c 1c 89 52 67 eb

1f 65 80 cc 3d b1 7f d6 78 9b 8d 5d 28 00 1c 3c 53 5a 01 d2 4b ba 6a 8f 97 25 93 93

5d ff dd 4a 95 64 06 cc ef da 51 a9 70 93 2d 0f 28 3b 63 62 81 13 05 c6 fd 54 9a 9c

c3 8c 07 e4

+Output Parameters:

f5\*\*: 248e86eba837

====================================================

6.8 Test set 6 [3GPP TS 35.232 V17.0.0 (2022-03)]

====================================================

+Intermediate Values:

IN when computing f5\*\* (first call of Keccak core):

51 7a ec 9c fa 6c 96 49 d1 5c 24 56 7f f4 ec 06 55 b6 9a 17 2a e2 2d c8 d6 fc 62 6c

f2 66 4a b0 e1 30 2e 31 4b 41 55 54 ef 8b 49 22 83 08 e3 b1 1f 65 de 8c c6 aa 70 c5

00 00 00 00 00 00 00 00 a5 5a 0e 77 15 9f c2 69 2b aa 26 44 5f 95 44 42 cd c9 89 f7

80 28 c8 89 c1 05 1d 88 56 ca 74 15 2f dc 58 d4 d9 4a 88 4c 1c b0 3a 8e 63 ac ab 83

75 e8 56 b5 61 ba 3a 06 25 e8 30 ac db 55 73 42 1f 00 00 00 00 00 00 80 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00

OUT when computing f5\*\* (first call of Keccak core):

f4 14 08 68 89 23 24 ea 27 07 2b a5 5d 0e 52 fb 0a 69 4e e9 3d 14 0c c8 5f 72 b1 29

2d 05 ca 77 3c c7 cf 7d 72 92 07 94 2d 24 bb 34 b8 68 c9 a9 0b 3e 4d 94 19 e6 f7 5a

ba c7 7d 91 73 cb 97 0a 56 aa ff 96 45 5a 27 25 c4 43 37 c7 11 00 70 71 ca f3 e4 e6

b1 45 cd 9a 98 bf a2 2e 2c a3 de eb 4e c7 a0 14 1e e4 7f e6 6d dd da e7 7e 38 53 e6

68 80 cb ba 5b ba 5f 5e c9 cc 45 f1 46 9b 44 2a f0 90 ea a8 62 e2 2b e5 23 38 16 d6

33 4c 72 96 d9 0c 23 f7 6c fa 99 24 fd 62 1a 58 e4 11 d4 55 f1 db 84 2f a6 7c 80 9b

26 29 51 ec 48 f4 86 0f 68 2f 10 f6 ad 96 57 b4 0b 33 ed 75 db ac 70 fb 4f bd c3 ef

5a e0 f9 d1

IN when computing f5\*\* (second call of Keccak core):

f4 14 08 68 89 23 24 ea 27 07 2b a5 5d 0e 52 fb 0a 69 4e e9 3d 14 0c c8 5f 72 b1 29

2d 05 ca 77 3c c7 cf 7d 72 92 07 94 2d 24 bb 34 b8 68 c9 a9 0b 3e 4d 94 19 e6 f7 5a

ba c7 7d 91 73 cb 97 0a 56 aa ff 96 45 5a 27 25 c4 43 37 c7 11 00 70 71 ca f3 e4 e6

b1 45 cd 9a 98 bf a2 2e 2c a3 de eb 4e c7 a0 14 1e e4 7f e6 6d dd da e7 7e 38 53 e6

68 80 cb ba 5b ba 5f 5e c9 cc 45 f1 46 9b 44 2a f0 90 ea a8 62 e2 2b e5 23 38 16 d6

33 4c 72 96 d9 0c 23 f7 6c fa 99 24 fd 62 1a 58 e4 11 d4 55 f1 db 84 2f a6 7c 80 9b

26 29 51 ec 48 f4 86 0f 68 2f 10 f6 ad 96 57 b4 0b 33 ed 75 db ac 70 fb 4f bd c3 ef

5a e0 f9 d1

OUT when computing f5\*\* (second call of Keccak core):

e9 6f 53 a0 20 5d 59 a9 c6 0b ba 4d 48 cf 00 fe 37 f3 73 ef d7 42 82 94 f1 9c c6 e4

82 e8 1d 9f 5a de 3c f5 b3 d6 f8 80 b9 cd 17 c7 07 8d c2 9b c1 ff a4 e0 fc fe 01 42

54 4e d7 8b 80 1f 51 79 31 9a 53 9d a2 5c 9e 36 70 42 88 43 82 f5 43 b9 6d fc 8b 20

b3 c5 f1 d6 5e 63 4a 14 55 8c bf 40 ae 86 78 d2 14 8e b3 03 60 16 b1 31 70 94 c5 3d

0c 82 b0 0a 39 13 2d f1 cf d5 5d a4 96 97 02 5b 2a 69 a0 00 53 de 6d 85 6b 97 bb 04

32 c1 dd 91 f9 21 2d fb b9 e1 5d 7d cc 87 38 02 5a bd b3 28 f1 40 59 41 17 93 c8 a7

11 dd c4 a0 97 b7 b8 81 30 48 9a 31 73 e0 62 23 79 9e 70 18 e1 18 84 29 f1 8c 45 f4

7a 2c df b4

+Output Parameters:

f5\*\*: 8e14d27886ae

Annex A (informative):
Change history

|  |
| --- |
| **Change history** |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 2025-03 | SA#107 | SP-250087 |  |  |  | Presented for information and approval | 1.0.0 |
| 2025-03 | SA#107 |  |  |  |  | Approved by TSG SA | 19.0.0 |
| 2025-07 | SA#108 | SP-250667 | 0001 | -  | D | Fixing a reference | 19.1.0 |