|  |
| --- |
| 3GPP TS 35.236 V0.2.0 (2024-08) |
| Technical Specification |
| 3rd Generation Partnership Project;Technical Specification Group Services and Security Aspects;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 3: Implementors’ Test Data and Design Conformance Test Data(Release 19) |
|   |
|  |  |
|  |
| ***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.© 2024, 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

Introduction 5

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 7

3.2 Symbols 8

3.3 Abbreviations 8

3.4 Radix 8

4 Structure of this specification 9

5 Implementors’ test data 9

5.1 MILENAGE-256-R 9

5.1.1 General 9

5.1.2 PRF based on Rijndael-256-256 9

6 Design conformance test data 11

6.1 Test data selection principles 11

6.2 Reference figure for identification of printed data 12

6.3 Milenage-256-R 12

# 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.

# Introduction

Editor's Note: This clause contains preface information provided by ETSI SAGE.

The present document contains a 256-bit example of set of algorithms, collectively called MILENAGE-256, which may be used as the authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*. It is not mandatory to use the particular algorithms specified in this document – all eight functions are operator-specifiable rather than being fully standardised. Operators electing to employ this example set can further personalise the algorithms (as described in the text).

The present document is one of four documents, which collectively comprise the entire specification of the example authentication and key generation algorithms. Namely:

- 3GPP TS 35.234 [2]: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 1: MILENAGE-256 General".

- 3GPP TS 35.235 [3]: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 2: MILENAGE-256 Algorithm Specification".

- **3GPP TS 35.236: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 3: Implementors’ Test and Design Conformance Test Data".**

- 3GPP TS 35.237 [4]: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 4: Summary and Results of Design and Evaluation".

# 1 Scope

Editor's Note: This clause contains scope information from ETSI SAGE for selected option.

The present document …

# 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 35.234: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 1: MILENAGE-256 General".

[3] 3GPP TS 35.235: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 2: MILENAGE-256 Algorithm Specification".

[4] 3GPP TS 35.237: "Specification of the MILENAGE-256 algorithm set: An example set of 256-bit 3GPP authentication and key generation functions f1, f1\*, f2, f2, f3, f5, f5, f5\* and f5\*\*; Document 4: Summary and Results of Design and Evaluation".

[5] 3GPP TS 33.102: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security Architecture".

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

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

[8] J. Daemen and V. Rijmen, "The design of Rijndael", Springer Verlag, 2002.

…

[x] <doctype> <#>[ ([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

# 3 Definitions of terms, symbols and abbreviations

Editor's Note: This clause contains notation that applies to the present document.

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP 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 3GPP TR 21.905 [1].

Definition format (Normal)

**<defined term>:** <definition>.

**example:** text used to clarify abstract rules by applying them literally.

**AKA-specific terminology**

**AMF:** Authentication Management Field

**AK:** Anonymity key

**AK\*:** Anonymity key used during resynchronisation

**CK:**  Cipher Key

**f1, f1\*, f2, f3, f4, f5, f5\*, f5\*\*:** Cryptographic functions used to derive AKA parameters

**IK:** Integrity Key

**K:** Subscriber key

**MAC-A:** Network Authentication Code

**MAC-S:** Resynchronisation Authentication Code

**RAND :** Random Challenge

**RES:** Response to Challenge

**SQN:** Sequence Number

**Additional terminology**

**AKSZ:** The length of the anonymity key **AK**, in octets

**ALGONAME:** An ASCII character string encoding of a name assigned for a particular instance/application of the MILENAGE-256 algorithm set instance

**𝑐0, 𝑐1, 𝑐2, 𝑐3, 𝑐4, 𝑐5, 𝑐6, 𝑐7:** 128-bit operator-customisable constants, used during the computation of **f1, f1\*, f2, f3, f4, f5, f5\***, and **f5\*\***

**CKSZ:** The length of the ciphering key response **CK**, in octets

**IN0, 𝐼𝑁1, 𝐼𝑁2, 𝐼𝑁3, 𝐼𝑁4, 𝐼𝑁5, 𝐼𝑁6, 𝐼𝑁7:** 256-bit instance-specific input values constructed within the computation of the functions **f1, f1\*, f2, f3, f4, f5, f5\***, and **f5\*\***

**IKSZ:** The length of the integrity key response **IK**, in octets

**KSZ:** The length of the subscriber key **K**, in octets

**MACSZ:** The length of the message authentication codes **MAC-A**, and **MAC-S**, in octets

**OP:** A 256-bit Operator Variant Algorithm Configuration Field that is a component of the functions **f1, f1\*, f2, f3, f4, f5, f5\*** and **f5\*\***

***OP*C:** A 256-bit value derived from *OP*, *ALGONAME*, *KSZ* and **K,** and used within the computations of the functions ***f1, f1\*, f2, f3, f4, f5, f5\**** and ***f5\*\****

**RESSZ:** The length of the response **RES**, in octets

**V:** A 256-bit intermediate value constructed from ALGONAME and KSZ, and used in the computation of OPC

NOTE: Bold variables above are part of the general AKA specification [8]. Additional explanation of the usage of boldface, italics, etc within MILENAGE-256 appears in the MILENAGE-256 Algorithm Specification [xx]. In the printout of test data, values are printed in courier typeface and indices are indicated by an underscore.

EXAMPLE: Values corresponding to KSZ are shown as K\_sz.

## 3.2 Symbols

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

Symbol format (EW)

<symbol> <Explanation>

= The assignment operator

:= The definition operator

⊕ The bitwise exclusive-OR operation

{ } Brackets are used to indicate a value given as a byte-array

EXAMPLE: *X* = { *X*[0] *X*[1] … *X*[*m*] }, where each *X*[*j*] is a byte

1A The *n*-bit binary value 00…001, given as a byte array, i.e. *X* = { *X*[0] … *X*[*m*] }, with *X*[0] = 1, and *X*[*j*] = 0 for *j* > 0

2A The *n*-bit binary value 00…010, given as a byte-array, i.e. *X* = { *X*[0] … *X*[*m*] }, with *X*[0] = 2, and *X*[*j*] = 0 for *j* > 0

∥ The concatenation of two byte arrays.

If *X* = { *X*[0] … *X*[*m*] } and *Y* = { *Y*[0] … *Y*[*n*] }, then

*X* || *Y* := { *X*[0] … *X*[*m*] *Y*[0] … *Y*[*n*]}.

AES-n AES with n-bit key (and 128-bit block size)

PRFK Pseudo-random function defined by key K

Rijndael-b-n Rijndael block cipher with b-bit block and n-bit key

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP 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 3GPP TR 21.905 [1].

Abbreviation format (EW)

<ABBREVIATION> <Expansion>

3GPP 3rd Generation Partnership Project

AES Advanced Encryption Standard

AKA Authentication and Key Agreement

ETSI SAGE ETSI Security Algorithms Group of Experts

MAC Message Authentication Code

MDPH Merkle-Damgård with Permutation and Hirose compression function

PRF Pseudo-Random Function

## 3.4 Radix

All test data print-outs below are given in hexadecimal notation, but omitting the usual "0x"-prefix.

EXAMPLE: A given value a3 corresponds to the decimal value 10\*16 + 3 = 163.

# 4 Structure of this specification

Editor's Note: this clause details how the present document is organized.

The test data provided in the present document comes in two categories. First, implementors' test data that can be used to verify the underlying block cipher component of MILENAGE-256, and test data for the PRF kernel constructed from block cipher. This includes also intermediate values produced during the computation of the kernel(s). This is sometimes refered to as "white-box" test data.

NOTE: For MILENAGE-256-R, the PRF-kernel is identical to the block cipher and thus only one common test-set is needed.

Input values for these test vectors were selected such that they change incrementally from one test to another, this strategy could be helpful for implementors to track possible errors in their implementation. There is also one test vector with pseudo random input values in order to catch other, unforeseen errors.

Next, conformance test data for the algorithm in the complete MILENAGE-256 algorithm set is provided, for different input and output parameter alternatives. This is sometimes refered to as "black-box" test data. The first test data for each input parameter set is more extensive and includes also intermediate values occurring during the computation.

This report is organised as follows:

* Clause 5 provides detailed test data for implementors, including also intermediate values occurring during computation, i.e. related to the cryptographic kernels (PRFs);
* Clause 6 provides design conformance data for MILENAGE-256-R. Principles for selecting test data is provided in sub-clause 6.1.

Source code for a reference implementation is provided in the Annex.

# 5 Implementors’ test data

Editor's Note: this clause provides implementors’ test data from ETSI SAGE.

## 5.1 MILENAGE-256-R

### 5.1.1 General

The kernel of MILENAGE-256-R is a PRF obtained directly as the Rijndael-256-256 block cipher. Therefore, test data for the PRF is identical to test data for Rijndael-256-256. Refer to the Algorithm Specification [3, clause 11] for details.

### 5.1.2 PRF based on Rijndael-256-256

=== PRF-RIJNDAEL-256-256 TEST #1 ===

KEY = { 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 }

X = { 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 }

RoundKey0 = { 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 | } |
| RoundKey1 | = | { | e2 | 63 | 63 | 63 | e2 | 63 | 63 | 63 | e2 | 63 | 63 | 63 | e2 | 63 | 63 | 63 |  |
|  |  |  | 98 | fb | fb | fb | 98 | fb | fb | fb | 98 | fb | fb | fb | 98 | fb | fb | fb | } |
| RoundKey2 | = | { | ef | 6c | 6c | 25 | 0d | 0f | 0f | 46 | ef | 6c | 6c | 25 | 0d | 0f | 0f | 46 |  |
|  |  |  | 4f | 8d | 8d | a1 | d7 | 76 | 76 | 5a | 4f | 8d | 8d | a1 | d7 | 76 | 76 | 5a | } |
| RoundKey3 | = | { | d3 | 54 | d2 | 2b | de | 5b | dd | 6d | 31 | 37 | b1 | 48 | 3c | 38 | be | 0e |  |
|  |  |  | a4 | 8a | 23 | 0a | 73 | fc | 55 | 50 | 3c | 71 | d8 | f1 | eb | 07 | ae | ab | } |
| RoundKey4 | = | { | 1e | b0 | b0 | c2 | c0 | eb | 6d | af | f1 | dc | dc | e7 | cd | e4 | 62 | e9 |  |
|  |  |  | 19 | e3 | 89 | 14 | 6a | 1f | dc | 44 | 56 | 6e | 04 | b5 | bd | 69 | aa | 1e | } |
| RoundKey5 | = | { | f7 | 1c | c2 | b8 | 37 | f7 | af | 17 | c6 | 2b | 73 | f0 | 0b | cf | 11 | 19 |  |
|  |  |  | 32 | 69 | 0b | c0 | 58 | 76 | d7 | 84 | 0e | 18 | d3 | 31 | b3 | 71 | 79 | 2f | } |
| RoundKey6 | = | { | 74 | aa | d7 | d5 | 43 | 5d | 78 | c2 | 85 | 76 | 0b | 32 | 8e | b9 | 1a | 2b |  |
|  |  |  | 2b | 3f | a9 | 31 | 73 | 49 | 7e | b5 | 7d | 51 | ad | 84 | ce | 20 | d4 | ab | } |
| RoundKey7 | = | { | 83 | e2 | b5 | 5e | c0 | bf | cd | 9c | 45 | c9 | c6 | ae | cb | 70 | dc | 85 |  |
|  |  |  | 34 | 6e | 2f | a6 | 47 | 27 | 51 | 13 | 3a | 76 | fc | 97 | f4 | 56 | 28 | 3c | } |
| RoundKey8 | = | { | b2 | d6 | 5e | e1 | 72 | 69 | 93 | 7d | 37 | a0 | 55 | d3 | fc | d0 | 89 | 56 |  |
|  |  |  | 84 | 1e | 88 | 17 | c3 | 39 | d9 | 04 | f9 | 4f | 25 | 93 | 0d | 19 | 0d | af | } |
| RoundKey9 | = | { | 7d | 01 | 27 | 36 | 0f | 68 | b4 | 4b | 38 | c8 | e1 | 98 | c4 | 18 | 68 | ce |  |
|  |  |  | 98 | b3 | cd | 9c | 5b | 8a | 14 | 98 | a2 | c5 | 31 | 0b | af | dc | 3c | a4 | } |
| RoundKey10 | = | { | cd | ea | 6e | 4f | c2 | 82 | da | 04 | fa | 4a | 3b | 9c | 3e | 52 | 53 | 52 |  |
|  |  |  | 2a | b3 | 20 | 9c | 71 | 39 | 34 | 04 | d3 | fc | 05 | 0f | 7c | 20 | 39 | ab | } |
| RoundKey11 | = | { | 16 | f8 | 0c | 5f | d4 | 7a | d6 | 5b | 2e | 30 | ed | c7 | 10 | 62 | be | 95 |  |
|  |  |  | e0 | 19 | 8e | b6 | 91 | 20 | ba | b2 | 42 | dc | bf | bd | 3e | fc | 86 | 16 | } |
| RoundKey12 | = | { | 7e | bc | 4b | ed | aa | c6 | 9d | b6 | 84 | f6 | 70 | 71 | 94 | 94 | ce | e4 |  |
|  |  |  | c2 | 3b | 05 | df | 53 | 1b | bf | 6d | 11 | c7 | 00 | d0 | 2f | 3b | 86 | c6 | } |
| RoundKey13 | = | { | 37 | f8 | ff | f8 | 9d | 3e | 62 | 4e | 19 | c8 | 12 | 3f | 8d | 5c | dc | db |  |
|  |  |  | 9f | 71 | 83 | 66 | cc | 6a | 3c | 0b | dd | ad | 3c | db | f2 | 96 | ba | 1d | } |
| RoundKey14 | = | { | ea | 0c | 5b | 71 | 77 | 32 | 39 | 3f | 6e | fa | 2b | 00 | e3 | a6 | f7 | db |  |
|  |  |  | 8e | 55 | eb | df | 42 | 3f | d7 | d4 | 9f | 92 | eb | 0f | 6d | 04 | 51 | 12 | } |
| Y | = | { | e6 | 2a | bc | e0 | 69 | 83 | 7b | 65 | 30 | 9b | e4 | ed | a2 | c0 | e1 | 49 |  |
|  |  |  | fe | 56 | c0 | 7b | 70 | 82 | d3 | 28 | 7f | 59 | 2c | 4a | 49 | 27 | a2 | 77 | } |
| === PRF-RIJNDAEL-256-256 TEST | #2 | === |  |  |  |  |  |  |  |  |  |  |
| KEY = { 01 01 01 01 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |  |
| 01 01 01 01 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | } |
| X = { 01 01 01 01 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 |  |
| 01 01 01 01 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | 01 | } |
| Y = { f6 f9 7c 67 72 | f2 | 04 | 88 | e3 | c0 | ee | c5 | 48 | 29 | 81 | b2 |  |
| bd 00 b1 5b bd | f9 | 40 | 06 | 9f | bf | 51 | 42 | ce | b3 | 96 | 88 | } |
| === PRF-RIJNDAEL-256-256 TEST #3 === |
| KEY = { 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 | } |
| X = { 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 | } |
| Y = { c6 22 | 7e | 77 40 | b7 | e5 | 3b | 5c | b7 | 78 | 65 | 27 | 8e | ab | 07 |  |
| 26 f6 | 23 | 66 d9 | aa | ba | d9 | 08 | 93 | 61 | 23 | a1 | fc | 8a | f3 | } |
| === RIJNDAEL-256-256KEY = { 00 00 | TEST #4 ===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 | } |
| X = { 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 | } |
| Y = { 15 9a | 08 e4 6e 61 | 6e | 6e | 99 | 78 | 50 | 20 | 10 | da | ff | 92 |  |
| 2e b3 | 62 e7 7d ca | af | 02 | ea | eb | 73 | 54 | eb | 8b | 8d | ba | } |
| === RIJNDAEL-256-256 | TEST #5 === |  |  |  |  |  |  |  |  |  |  |  |
| KEY = { 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 | } |
| X = { 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 | 01 | } |
| Y = { 4e 76 | ca 69 96 71 | 25 | a9 | 63 | 6f | 35 | 54 | 22 | 95 | 56 | f6 |  |
| e2 b2 | 35 1c b4 fd | 10 | b4 | e0 | 52 | af | d8 | 5b | eb | df | a8 | } |

=== PRF-RIJNDAEL-256-256 TEST #6 ===

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| KEY = { ff ff ffff ff ffX = { ff ff ffff ff ff | ff ffff ffff ffff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | ffff ff ff | }} |
| Y = { f3 6c b6 | c7 a7 | 57 | 2f | 19 | 30 | 7a | 31 | e4 | ec | 4c | a4 | c8 |  |
| 2d 27 31 | fb 21 | f5 | 9c | af | 13 | 3f | e8 | 16 | a5 | 44 | 24 | a5 | } |
| === PRF-RIJNDAEL-256-256 TEST | #7 | === |  |  |  |  |  |  |  |  |  |  |
| KEY = { 00 01 02 03 04 | 05 | 06 | 07 | 08 | 09 | 0a | 0b | 0c | 0d | 0e | 0f |  |
| 10 11 12 13 14 | 15 | 16 | 17 | 18 | 19 | 1a | 1b | 1c | 1d | 1e | 1f | } |
| X = { 00 11 22 33 44 | 55 | 66 | 77 | 88 | 99 | aa | bb | cc | dd | ee | ff |  |
| 10 21 32 43 54 | 65 | 76 | 87 | 98 | a9 | ba | cb | dc | ed | fe | 0f | } |
| Y = { 28 8f a9 d2 3d | 00 | d9 | dc | 0a | 39 | b3 | 3f | a9 | 28 | 67 | c6 |  |
| 48 8b 5e 0f 18 | a6 | f7 | 4c | 07 | 20 | 78 | ec | 81 | 54 | 62 | e6 | } |
| === PRF-RIJNDAEL-256-256 TEST KEY = { 0f 3a 40 4e b7 | #8f3 | === 49 | d4 | 7b | e0 | 0f | 1c | 19 | df | c9 | 0a |  |
| 3a 04 0a 79 96 | 30 | fe | 74 | 5f | e5 | f0 | a9 | dd | 58 | 10 | 1a | } |
| X = { 56 c8 74 16 30 | 0f | 5c | 68 | 59 | 77 | ba | ce | e7 | ce | f2 | ad |  |
| fc bd de 02 ed | 55 | 15 | 9c | a3 | eb | 6d | 89 | 41 | 26 | be | ce | } |
| Y = { a5 f0 c2 bb a8 | b1 | 31 | cb | 82 | ea | 65 | 4e | ba | b1 | 52 | 21 |  |
| 3d f5 0b d1 84 | f4 | 87 | b2 | 2d | 5a | ea | b8 | 7c | c3 | b0 | 24 | } |

# 6 Design conformance test data

Editor's Note: this clause provides background information from ETSI SAGE.

## 6.1 Test data selection principles

##### The overall MILENAGE-256 construct has a huge number of possible instantiations based on exact choices of the parameter sizes (KEYsz, RESsz, …), and it is impossible to provide test vectors for each combination. The strategy has instead been to provide test vectors for a few selections that are, with high confidence, providing assurance in the correctness of an implementation.

##### There are five main test-sets, each first selecting the different inputs, i.e. the input parameter sizes, the input parameter values, and values for the operator configurable constants, having the following structure:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Set | *KEYsz* | *RANDsz* | *SQNsz* | *c0..c7* | *KEY/OP/RAND/ SQN/AMF* | Set characterisation |
| #1 | 32 | 32 | 12 | All distinct | All distinct | Maximum |
| #2 | 16 | 16 | 6 | All distinct | All distinct | Minimum |
| #3 | 32 | 22 | 9 | All ones | All ones | Odd |
| #4 | 32 | 16 | 6 | Default values | All random | Expected |
| #5 | 32 | 32 | 6 | All zeroes | All random | Desired |

##### Then, for each of the above input selections, the following five sub-tests, selecting the five output parameters sizes, are provided as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subset | *RESsz* | *CKsz* | *IKsz* | *MACsz* | *AKsz* | Set characterisation |
| a | 32 | 32 | 32 | 32 | 12 | Maximum (includes also debug info) |
| b | 4 | 16 | 16 | 8 | 6 | Minimum |
| c | 7 | 29 | 17 | 23 | 9 | Odd |
| d | 8 | 32 | 32 | 8 | 6 | Expected |
| e | 32 | 32 | 32 | 16 | 6 | Desired |

##### The sets "Maximum" and "Minimum" adopt the maximum and minimum allowed sizes of the corresponding parameters, respectively. Sets "Odd" check borderline unusual sizes. Sets "Expected" and "Desired" are two variants of what ETSI SAGE anticipates could likely be used in practice.

##### Thus, there are in total 25 test vectors, where five of them, the tests #{1-5}a, additionally include details of intermediate values for debugging purposes.

## 6.2 Reference figure for identification of printed data

##### For the purpose of this clause, the labels employed in the printed data below can be identified by reference to the following figure.



Figure 6.2-1: Reference figure for identification of printed data. For simplicity, only *f2* is shown.

##### In all cases, data has been produced using ALGONAME = "MILENAGE2.0" as defined in the Algorithm Specification [2].

##### NOTE: Some combinations of parameter-values will most likely not be encountered in practice, such as values with SQNsz ≠ AKsz, since these two parameters are typically identical. Nevertheless, such values are provided below for completeness.

## 6.3 Milenage-256-R

##### Tests #{1-5}a contain intermediate values and may be used for debugging purposes, other tests #{1- 5}{b-e} can be seen as conformance tests without intermediate values.

Editor's Note: to complete with data from ETSI SAGE.

Annex A (informative):
Reference implementation (C/C++)

Editor's Note: this clause provides an informal C/C++ implementation example of Milenage-256 defined by ETSI SAGE.

Annex B (informative):
Change history

|  |
| --- |
| **Change history** |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2024-02 | SA3#115 | S3-240405 |  |  |  | TS skeleton | 0.0.0 |
| 2024-02 | SA3#115 | S3-240819 |  |  |  | TS skeleton using 3GPP template | 0.0.1 |
| 2024-02 | SA3#115 | S3-240409 |  |  |  | Addition of introduction | 0.1.0 |
| 2024-08 | SA3#117 | S3-243424 |  |  |  | Addition of the text based on the selection of Milenage-256-R to specify Milenage-256 algorithm. | 0.2.0 |