ETSI/SAGE	**Version: 1.0**
Specification	**Date: 18th June 2010**

Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3

Document 3: Implementors’ Test Data

Blank Page

PREFACE

This specification has been prepared by the 3GPP Task Force, and gives detailed test data for implementors of the algorithm set. It provides visibility of the internal state of the ZUC algorithm to aid in its realization, as well as black box input-output test data. The test data has been selected to give a high degree of confidence that the implementation is correct. However, no claim is made that conformance with this test data guarantees a correct implementation. This document is the third of three, which between them form the entire specification of the 3GPP Confidentiality and Integrity Algorithms:

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
  Document 1: ***128-EEA3*** and ***128-EIA3*** Specifications.

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
  Document 2: **ZUC** Specification.

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
  Document 3: Implementors' Test Data.

This document is purely informative. The normative part of the specification of the ***128-EEA3*** (confidentiality) and the ***128-EIA3*** (integrity) algorithms is in the main body of document 1. The normative part of the specification of **ZUC** is found in document 2.

Blank Page

TABLE OF CONTENTS

1 OUTLINE OF THE DESIGN CONFORMANCE TEST DATA 8

2 INTRODUCTORY INFORMATION 8

2.1 Introduction 8

2.2 Radix 8

2.3 Bit/Byte ordering 8

2.4 Presentation of input/output data 8

2.5 Coverage 8

3 ZUC 9

3.1 Overview 9

3.2 Format 9

3.3 Test Set 1 9

3.4 Test Set 2 10

3.5 Test Set 3 12

3.6 Test Set 4 13

4 CONFIDENTIALITY ALGORITHM 128-*EEA3* 14

4.1 Overview 14

4.2 Format 14

4.3 Test Set 1 14

4.4 Test Set 2 15

4.5 Test Set 3 15

4.6 Test Set 4 16

4.7 Test Set 5 17

5 INTEGRITY ALORITHM 128-*EIA3* 18

5.1 Overview 18

5.2 Test Set 1 18

5.3 Test Set 2 19

5.4 Test Set 3 19

5.5 Test Set 4 19

5.6 Test Set 5 20

REFERENCES

[1] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (3G TS 33.102 version 6.3.0)

[2] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Cryptographic Algorithm Requirements; (3G TS 33.105 version 6.0.0)

[3] Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
Document 1: ***128-EEA3*** and ***128-EIA3*** specifications.

[4] Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
Document 2: **ZUC** specification.

[5] Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
Document 3: Implementors' Test Data.

[6] Specification of the 3GPP Confidentiality and Integrity Algorithms ***128-EEA3*** & ***128-EIA3***.
Document 4: Design and Evaluation Report.

1 OUTLINE OF THE TEST DATA

Section 2 introduces the algorithms and describes the notation used in the subsequent sections.

Section 3 provides test data for the **ZUC** algorithm.

Section 4 provides test data for the confidentiality algorithm ***128-EEA3***.

Section 5 provides test data for the integrity algorithm ***128-EIA3***.

2 INTRODUCTORY INFORMATION

2.1 Introduction

Within the security architecture of the 3GPP system there are two standardised algorithms; a confidentiality algorithm ***128-EEA3***, and an integrity algorithm ***128-EIA3***. These algorithms are specified in a companion document[3].

This document provides sets of input/output test data for 'black box' testing of physical realisations of the **ZUC**, ***128-EEA3*** and ***128-EIA3*** algorithms. It also provides some internal values within the ZUC algorithm, to aid in its implementation.

2.2 Radix

Unless stated otherwise, all test data values presented in this document are in hexadecimal.

2.3 Bit/Byte ordering

All data variables in this specification are presented with the most significant bit (or byte) on the left hand side and the least significant bit (or byte) on the right hand side.

2.4 Presentation of input/output data

The basic data processed by the ***128-EEA3*** and ***128-EIA3*** algorithms are bit streams. In general in this document the data is presented in hexadecimal format as bytes, thus the last byte shown as part of an input or output data stream may include between 0 and 7 bits that are ignored once the **LENGTH** parameter is taken into account. (The least significant bits of the byte are ignored).

2.5 Coverage

For each of the algorithms the test data have been selected such that, provided the entire test set is run:

- Each key bit will have been in both the '1' and the '0' states,
- Each bit of the initialisation fields (COUNT, FRESH, BEARER, DIRECTION) will have been in both the '1' and the '0' states,
- If the test set for ***128-EEA3*** is run every entry in the internal S-boxes S_0 and S_1 will have been accessed.

3 ZUC

3.1 Overview

The test data sets presented here are for the ZUC stream cipher algorithm.

3.2 Format

Each test set starts by showing the input and output data values.

This is followed by a table showing the state of the LFSR at the beginning of the computation.

Then for the first 10 steps of the initialisation the content of X_0, X_1, X_2, X_3, R_1, R_2 is given in a table.

Then the state of the LFSR and the nonlinear function F at the end of the initialization is given.

For the first 3 steps of keystream generation X_0, X_1, X_2, X_3, R_1, R_2 are given in a table.

3.3 Test Set 1

input:

Key: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

IV: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

output:

z_1: b8766c51

z_2: d9d6e338

Initialisation Mode

LFSR-state at the beginning:

i	S_{0+i}	S_{1+i}	S_{2+i}	S_{3+i}	S_{4+i}	S_{5+i}	S_{6+i}	S_{7+i}
0	0044d700	0026bc00	00626b00	00135e00	00578900	0035e200	00713500	0009af00
8	004d7800	002f1300	006bc400	001af100	005e2600	003c4d00	00789a00	0047ac00

i	X_0	X_1	X_2	X_3	R_1	R_2
0	008f9a00	f100005e	af00006b	6b000089	67822141	62a3a55f
1	e1b8ac00	260000d7	780000e2	5e00004d	474a2e7e	119e94bb
2	dffb335f	4d000035	13000013	890000c4	c29687a5	e9b6eb51

3	1992a541	9a0000bc	c400009a	E2000026	29c272f3	8cac7f5d
4	4b185941	ac000078	f100005e	350000af	2c85a655	24259cb0
5	24d5398a	335f00f1	260000d7	af00006b	cbfbc5c0	1af25e87
6	77cac388	a541008f	4d000035	780000e2	9b442b52	dd05d8f9
7	edbe212f	5941e1b8	9a0000bc	13000013	d1454bbb	9a05c00c
8	4b6b798a	398adffb	ac000078	C400009a	b1467784	f86cb302
9	445c83e0	c3881992	335f00f1	F100005e	bf623a76	5bf3c609

LFSR-state after completion of the initialisation mode:

i	S_{0+i}	S_{1+i}	S_{2+i}	S_{3+i}	S_{4+i}	S_{5+i}	S_{6+i}	S_{7+i}
0	602167c7	13ddb700	28d57643	73972d1a	487354b9	2c6cdc4c	067dd77a	6e20e6fd
8	0237b3f7	11eae23e	3c323c40	25c0051c	0eebe0aa	0ac7ca6c	1b25af1b	017cdf7d

FSM-state after completion of the initialisation mode:

R_1 = b6caf723

R_2 = 759e5bbe

Keystream mode

i	X_0	X_1	X_2	X_3	R_1	R_2
0	02f9af1b	051c23d5	e6fd58d9	7643c042	9cef8659	660bf2c6
1	b38fdf7d	e0aa7864	b3f70cfb	2d1a27bb	0d8c6935	2292146a
2	6751f71d	ca6c4b80	e23edc41	54b951aa	9017a50a	3dbf77b5

3.4 Test Set 2

input:

Key: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

IV: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff

output:

z_1: 9140c243

z_2: 1419ef9d

Initialisation Mode

LFSR-state at the beginning:

i	S_{0+i}	S_{1+i}	S_{2+i}	S_{3+i}	S_{4+i}	S_{5+i}	S_{6+i}	S_{7+i}
0	7fc4d7ff	7fa6bcff	7fe26bff	7f935eff	7fd789ff	7fb5e2ff	7ff135ff	7f89afff
8	7fcd78ff	7faf13ff	7febc4ff	7f9af1ff	7fde26ff	7fbc4dff	7ff89aff	7fc7acff

i	X_0	X_1	X_2	X_3	R_1	R_2
0	ff8f9aff	f1ffff5e	afffff6b	6bffff89	b51c2110	30a3629a
1	7a49acff	26ffffd7	78ffffe2	5effff4d	a75b6f4b	1a079628
2	37effe21	4dffff35	13ffff13	89ffffc4	9810b315	99296735
3	ea7ad09d	9affffbc	c4ffff9a	E2ffff26	4c5bd8eb	2d577790
4	463410ed	acffff78	f1ffff5e	35ffffaf	a13dcb66	21d0939f
5	073433d4	fe21fff1	26ffffd7	afffff6b	cc5ce260	cb2c6e6a
6	8b9974d3	d09dff8f	4dffff35	78ffffe2	7bf31296	66fdda3d
7	1ec99ac6	10ed7a49	9affffbc	13ffff13	f89270a0	a0d624e2
8	df58b392	33d437ef	acffff78	C4ffff9a	ace859f9	92f2bf2c
9	9f4e71ef	74d3ea7a	fe21fff1	F1ffff5e	453b8d43	94ed1a32

LFSR-state after completion of the initialisation mode:

i	S_{0+i}	S_{1+i}	S_{2+i}	S_{3+i}	S_{4+i}	S_{5+i}	S_{6+i}	S_{7+i}
0	55ac079e	057c2f4a	7cd17391	59f72734	5410ea9c	334a2fac	3a8a1307	5e6fd3c0
8	15296765	234ae453	16ba2dbe	78c3ca11	270a4e8a	09de7803	17f9d663	41bc1070

FSM-state after completion of the initialisation mode:

R_1 = b8a8de56

R_2 = 032f1ba6

Keystream mode

i	X_0	X_1	X_2	X_3	R_1	R_2
0	8378d663	ca114695	d3c06694	7391ab58	6489a4de	e2e7140d

1	b7041070	4e8a2d74	67657514	27340af8	ed35f777	04645ee0
2	17154028	7803f187	e453bcdf	ea9cf9a2	33c7777f	f87cda51

3.5 Test Set 3

input:

Key: 3d 4c 4b e9 6a 82 fd ae b5 8f 64 1d b1 7b 45 5b

IV: 84 31 9a a8 de 69 15 ca 1f 6b da 6b fb d8 c7 66

output:

z_1: 9a579b30

z_2: ecbe5f34

Initialisation Mode

LFSR-state at the beginning:

i	S_{0+i}	S_{1+i}	S_{2+i}	S_{3+i}	S_{4+i}	S_{5+i}	S_{6+i}	S_{7+i}
0	1ec4d784	2626bc31	25e26b9a	74935ea8	355789de	4135e269	7ef13515	5709afca
8	5acd781f	47af136b	326bc4da	0e9af16b	58de26fb	3dbc4dd8	22f89ac7	2dc7ac66

i	X_0	X_1	X_2	X_3	R_1	R_2
0	5b8f9ac7	f16b8f5e	afca826b	6b9a3d89	9c62829f	5df00831
1	2d72ac66	26fb64d7	781ffde2	5ea84c4d	3d533f3a	80ff1faf
2	937c15fa	4dd81d35	136bae13	89de4bc4	2ca57e9d	d1db72f9
3	22f46022	9ac7b1bc	c4dab59a	E269e926	0e8dc40f	60921a4f
4	f1a21ca6	ac667b78	f16b8f5e	35156aaf	16c81467	da8e7d8a
5	a2587876	15fa45f1	26fb64d7	afca826b	50c9eaa4	754debd7
6	1b4053c2	60225b8f	4dd81d35	781ffde2	05fd8a8b	aa32c96d
7	e3fe7f36	1ca62d72	9ac7b1bc	136bae13	94903322	52919977
8	e1001bd7	7876937c	ac667b78	C4dab59a	f286586f	624edb3d
9	de1d7c83	53c222f4	15fa45f1	F16b8f5e	4bb9a070	a0978960

LFSR-state after completion of the initialisation mode:

i	S_{0+i}	S_{1+i}	S_{2+i}	S_{3+i}	S_{4+i}	S_{5+i}	S_{6+i}	S_{7+i}
0	71d95526	6840e635	2a654543	7b6b3c5a	0acd2958	3e4eaa19	6bb7b0f1	3f63bd17
8	4215a5bd	42ac9e4f	6598feee	136f9d9d	3f85c880	30a587bb	247efa23	0cd20507

FSM-state after completion of the initialisation mode:

R_1 = 297a501f

R_2 = 00d6b0bd

Keystream mode

i	X_0	X_1	X_2	X_3	R_1	R_2
0	19a4fa23	9d9d8559	bd177c9d	4543e3b2	4fde4a74	b733fc3e
1	a1070507	c880cb31	a5bdd76f	3c5ad081	32a4f017	83dd90be
2	70ac8b57	87bb26df	9e4f7ec7	295854ca	4b479636	51e21b60

3.6 Test Set 4

input:

Key: 4d 32 0b fa d4 c2 85 bf d6 b8 bd 00 f3 9d 8b 41

IV: 52 95 9d ab a0 bf 17 6e ce 2d c3 15 04 9e b5 74

output:

z_1: c27a7b79

z_2: 8b9759ae

……

Z_{2000}: e2459b19

4 CONFIDENTIALITY ALGORITHM *128-EEA3*

4.1 Overview

The test data sets presented here are for the ***128-EEA3*** confidentiality algorithm.

4.2 Format

Each test set shows the various inputs to the algorithm including the plain text data stream to be encrypted/decrypted. (The length field is in decimal).

The fields are:

Key	= CK[0] || … || CK[127]
Count	= COUNT-C[0] || … || COUNT-C[31]
Bearer	= BEARER[0] || … || BEARER[4]
Direction	= DIRECTION[0]
Length	= Length of data in decimal
Plaintext	= PT[0] || PT[1] || … || PT[Length-1]

This is followed by the modified input data, i.e. it is the bitwise exclusive-or of the corresponding keystream and the input data to the algorithm.

Ciphertext = CT[0] || CT[1] || … || CT[Length-1]

As this is a bitwise stream cipher it is purely a matter of context whether the operation is regarded as “encryption” or “decryption”. For the purposes of this document we regard the input as Plaintext and the output as Ciphertext.

The first test set is shown twice, once in binary format, once in hexadecimal format. This is to explicitly show the relationship between the binary data and the hexadecimal presentation.

The remainder of the test sets are presented in hexadecimal format only.

4.3 Test Set 1

```
Key       = (hex) 17 3d 14 ba 50 03 73 1d 7a 60 04 94 70 f0 0a 29

Key        = (bin) 00010111 00111101 00010100 10111010 01010000 00000011 01110011 00011101

                   01111010 01100000 00000100 10010100 01110000 11110000 00001010 00101001

Count     = (hex) 66035492

Count     = (bin) 01100110 00000011 01010100 10010010

Bearer    = (hex) f

Bearer    = (bin) 1111

Direction = (hex) 0

Direction = (bin) 0

Length    = 193 bits

Plaintext:
```

(hex) 6cf65340 735552ab 0c9752fa 6f9025fe 0bd675d9 005875b2 00000000

(bin) 01101100 11110110 01010011 01000000 01110011 01010101 01010010 10101011

00001100 10010111 01010010 11111010 01101111 10010000 00100101 11111110

00001011 11010110 01110101 11011001 00000000 01011000 01110101 10110010

0

Ciphertext:

(hex) f4132a26 e5977925 50db26b5 34c37b1d c765398e 1beaf2a3 80000000

(bin) 11110100 00010011 00101010 00100110 11100101 10010111 01111001 00100101

01010000 11011011 00100110 10110101 00110100 11000011 01111011 00011101

11000111 01100101 00111001 10001110 00011011 11101010 11110010 10100011

1

4.4 Test Set 2

Key = e5 bd 3e a0 eb 55 ad e8 66 c6 ac 58 bd 54 30 2a

Count = 56823

Bearer = 18

Direction = 1

Length = 800 bits

Plaintext:

14a8ef69 3d678507 bbe7270a 7f67ff50 06c3525b 9807e467 c4e56000 ba338f5d 42955903
67518222 46c80d3b 38f07f4b e2d8ff58 05f51322 29bde93b bbdcaf38 2bf1ee97 2fbf9977
bada8945 847a2a6c 9ad34a66 7554e04d 1f7fa2c3 3241bd8f 01ba220d

Ciphertext:

48cbab83 c9d37b6b 7900f402 f6705bc6 f4defc54 28fe6cd0 4f096a5c 1d96b7e0 8c2caefd
5a5e808f 92ab8e8c 5e9c0189 e8b90e98 3a7288a0 848ad04c 19e5045f 3017a7ee 6ae4c583
f811f214 be598848 ddde8313 f2e38bc1 fb0a2f56 a04bf066 9df1708f

4.5 Test Set 3

Key = d4 55 2a 8f d6 e6 1c c8 1a 20 09 14 1a 29 c1 0b

Count = 76452ec1

Bearer = 2

Direction = 1

```
Length    = 1570 bits

Plaintext:

     38f07f4b e2d8ff58 05f51322 29bde93b bbdcaf38 2bf1ee97 2fbf9977 bada8945 847a2a6c
     9ad34a66 7554e04d 1f7fa2c3 3241bd8f 01ba220d 3ca4ec41 e074595f 54ae2b45 4fd97143
     20436019 65cca85c 2417ed6c bec3bada 84fc8a57 9aea7837 b0271177 242a64dc 0a9de71a
     8edee86c a3d47d03 3d6bf539 804eca86 c584a905 2de46ad3 fced6554 3bd90207 372b27af
     b79234f5 ff43ea87 0820e2c2 b78a8aae 61cce52a 0515e348 d196664a 3456b182 a07c406e
     4a207912 71cfeda1 65d535ec 5ea2d4df 40000000

Ciphertext:

    6a1186f2 383fa7bd e1c6db83 02874009 df47e0e3 b13d80da b899fef6 afd696b7 7d016fbe
    e04741ce eb83c0e5 61dc6b74 cc871f5e 9a0e2288 c19b5eaf efc4be84 35876114 572e80be
    10d174b4 43b1d543 7dd85778 263055e6 eb570430 2409e19b 170bbebf ee767ae9 c7381ea2
    0ca9c5a6 3feb7276 a727adb5 a296f9ec 11f2463d e655ecf6 4dc743f7 25f959fe 99b1e59c
    5aa0ee65 878efea6 e02f7668 88c64098 8b9575c0 72d89745 4c6ad19f 6b8991ae 7e979ae6
    02c4c072 96c192af d9b698a1 7446f191 00000000
```

4.6 Test Set 4

```
Key       = db 84 b4 fb cc da 56 3b 66 22 7b fe 45 6f 0f 77

Count     = e4850fe1

Bearer    = 10

Direction = 1

Length    = 2798 bits

Plaintext:

     e539f3b8 973240da 03f2b8aa 05ee0a00 dbafc0e1 82055dfe 3d7383d9 2cef40e9 2928605d
     52d05f4f 9018a1f1 89ae3997 ce19155f b1221db8 bb0951a8 53ad852c e16cff07 382c93a1
     57de00dd b125c753 9fd85045 e4ee07e0 c43f9e9d 6f414fc4 d1c62917 813f74c0 0fc83f3e
     2ed7c45b a5835264 b43e0b20 afda6b30 53bfb642 3b7fce25 479ff5f1 39dd9b5b 995558e2
     a56be18d d581cd01 7c735e6f 0d0d97c4 ddc1d1da 70c6db4a 12cc9277 8e2fbbd6 f3ba52af
     91c9c6b6 4e8da4f7 a2c266d0 2d001753 df089603 93c5d568 88bf49eb 5c16d9a8 0427a416
     bcb597df 5bfe6f13 890a07ee 1340e647 6b0d9aa8 f822ab0f d1ab0d20 4f40b7ce 6f2e136e
     b67485e5 07804d50 4588ad37 ffd81656 8b2dc403 11dfb654 cdead47e 2385c343 6203dd83
     6f9c64d9 7462ad5d fa63b5cf e08acb95 32866f5c a787566f ca93e6b1 693ee15c f6f7a2d6
     89d97417 98dc1c23 8e1be650 733b18fb 34ff880e 16bbd21b 47ac0000

Ciphertext:

     fb00b11e d9b1c865 fca684fa 2a96ca36 ce5a06de 43d2d9d4 05ab0905 a46bb943 1ce16e76
     f971abdd fd2b64c7 a493442e da106b61 7f4f90d9 93bc06e5 912b932f fd406ab5 1e130bcf
     f59f8350 eee4b090 163fd32d 8ce8a7ec abf1f823 c579b3e3 ce89ebf7 94e0d039 a102e4e0
     b3344969 0796433e 368cf535 d0c1e6ff ef714e34 d8bdd852 4ebe3f7d 31291820 4f989e8c
     a988ff03 93aa7ab0 e9cbd933 ffb84fc7 bf243f78 e9984c33 9263efd9 93765dfb 47f2b4d7
     4a667a79 05130ef5 e2753873 86e0b5ef abbd4aff be45ca35 c07fc006 e52526aa 319e290c
     fb81ce1d 6c8b2a6e 09242f47 da343b5d c7232253 0cc1bb9a e20f61c3 574ff446 4a876225
     61b4bad8 fd3eec8c 7c66afa6 e93181b3 a681ba51 7bdfc384 17083cff 383216db cd6d77a7
     541cf0e0 822dcf96 0fde2ed0 7f2ac67a 37e78c9f 1bae8296 f5e7e87f 0ecc93dd cf0be6c1
     c9080718 8f314cc9 f419edfd 18bcfc1a 998be4fc e13ab076 b8700000
```

4.7 Test Set 5

```
Key       = e1 3f ed 21 b4 6e 4e 7e c3 12 53 b2 bb 17 b3 e0

Count     = 2738cdaa

Bearer    = 1a

Direction = 0

Length    = 4019 bits

Plaintext:

     8d74e20d 54894e06 d3cb13cb 3933065e 8674be62 adb1c72b 3a646965 ab63cb7b 7854dfdc
     27e84929 f49c64b8 72a490b1 3f957b64 827e71f4 1fbd4269 a42c97f8 24537027 f86e9f4a
     d82d1df4 51690fdd 98b6d03f 3a0ebe3a 312d6b84 0ba5a182 0b2a2c97 09c090d2 45ed267c
     f845ae41 fa975d33 33ac3009 fd40eba9 eb5b8857 14b768b6 97138baf 21380eca 49f644d4
     8689e421 5760b906 739f0d2b 3f091133 ca15d981 cbe401ba f72d05ac e05cccb2 d297f4ef
     6a5f58d9 1246cfa7 7215b892 ab441d52 78452795 ccb7f5d7 9057a1c4 f77f80d4 6db2033c
     b79bedf8 e60551ce 10c667f6 2a97abaf abbcd677 2018df96 a282ea73 7ce2cb33 1211f60d
     5354ce78 f9918d9c 206ca042 c9b62387 dd709604 a50af16d 8d35a890 6be484cf 2e74a928
     99403643 53249b27 b4c9ae29 eddfc7da 6418791a 4e7baa06 60fa6451 1f2d685c c3a5ff70
     e0d2b742 92e3b8a0 cd6b04b1 c790b8ea d2703708 540dea2f c09c3da7 70f65449 e84d817a
     4f551055 e19ab850 18a0028b 71a144d9 6791e9a3 57793350 4eee0060 340c69d2 74e1bf9d
     805dcbcc 1a6faa97 6800b6ff 2b671dc4 63652fa8 a33ee509 74c1c21b e01eabb2 16743026
     9d72ee51 1c9dde30 797c9a25 d86ce74f 5b961be5 fdfb6807 814039e7 137636bd 1d7fa9e0
     9efd2007 505906a5 ac45dfde ed7757bb ee745749 c2963335 0bee0ea6 f409df45 80160000

Ciphertext:

    cec07e88 aab38609 4b7dba70 a418a25b 322341e3 7b2f4df3 465b06d9 a9cc8db2 04d38286
    86612ee2 c9941ceb 50de2e19 366112e6 89a21153 1088c312 dcf62d09 dfe154d0 1d39a32b
    f1c9fe06 14ca6da6 2be1f544 5685cc37 47e4aeff 7a307837 91e488bd c62ca020 489ff3fb
    8171fe65 0c81d7c4 5751d1c9 5508fade 77667da9 f596c6a6 c7c896b6 33e5bcbc e489f1c1
    5d988d0c cb6a914a 0f17f360 b2224d19 928dbca5 618da0f2 e40441c0 4d91a729 eae9f398
    26e8e9ac 89b7fad2 4fbfbf44 a406d614 64680338 7c3215ad c2421660 7f0428dc f91a6b69
    283089bf 9d993654 47cdbb44 2a9ecead a4aaa362 5a7b25fd 0307c704 0e770dac 7d33c816
    a9c77228 622447c8 1b191622 0d72c707 e8ba50dc d5ab9440 82a7224a a41dfd5c 45355dd1
    fdd731b6 008b898f 34a606ec c58b0953 733d7ccb 3c52c70c cdf54ee1 a10f52cd eb6572f2
    741920e4 09482d48 10819eae 547676d2 c6f9765a fe1f3cec 664114f9 59c4150c c30f698f
    44df502f 6e03ccd6 b6fe7883 9c7e9da1 0afed885 bc7dd524 57ed4072 86e24b83 de0a03c6
    ac08d01b f30be95e 52a49d24 0e76a49a c5832e13 26534e22 e20e3c75 04d3e4d6 575f4093
    1e2dbe17 df32d01d f6109acc 26bb8250 4e42f1e8 d2b05436 a49e0396 51ad91a2 51c90f9b
    354b9545 dad2fa51 e1ac01d5 7b7702d7 b74bf810 34385a77 1fdbde37 dcad7d46 835e4000
```

5 INTEGRITY ALORITHM *128-EIA3*

5.1 Overview

The test data sets presented here are for the ***128-EIA3*** integrity algorithm.

Each test set shows the various inputs to the algorithm including the plain text data stream to be 'MAC'd. The length field is in decimal.

The fields are:

Key	= IK[0] || … || IK[127]
Count	= COUNT-I[0] || … || COUNT-I[31]
Bearer	= BEARER[0] || … || BEARER[4]
Direction	= DIRECTION[0]
Length	= Length of data in decimal
Message	= MESSAGE[0] || … || MESSAGE[Length-1]

This is followed by the calculated value for MAC-I.

Output = MAC-I[0] || … || MAC-I[31]

The first test set is shown twice, once in binary format, once in hexadecimal format. This is to explicitly show the relationship between the binary data and the hexadecimal presentation.

The remainder of the test sets are presented in hexadecimal format only.

5.2 Test Set 1

```
Key       = (hex) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Key       = (hex) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

                  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Count     = (hex) 0

Count     = (bin) 0

Bearer    = (hex) 0

Bearer    = (bin) 0

Direction = (hex) 0

Direction = (bin) 0

Length    = 1 bits

Message:

     (hex) 00000000

     (bin) 0

MAC: (hex) c3411ed2
```

(bin) 11000011 01000001 00011110 11010010

5.3 Test Set 2

Key = 47 05 41 25 56 1e b2 dd a9 40 59 da 05 09 78 50

Count = 561eb2dd

Bearer = 14

Direction = 0

Length = 90 bits

Message:

00000000 00000000 00000000

MAC: 61f6b94e

5.4 Test Set 3

Key = c9 e6 ce c4 60 7c 72 db 00 0a ef a8 83 85 ab 0a

Count = a94059da

Bearer = a

Direction = 1

Length = 577 bits

Message:

983b41d4 7d780c9e 1ad11d7e b70391b1 de0b35da 2dc62f83 e7b78d63 06ca0ea0 7e941b7b
e91348f9 fcb170e2 217fecd9 7f9f68ad b16e5d7d 21e569d2 80ed775c ebde3f40 93c53881
00000000

MAC: f99b0330

5.5 Test Set 4

Key = c8 a4 82 62 d0 c2 e2 ba c4 b9 6e f7 7e 80 ca 59

Count = 05097850

Bearer = 10

Direction = 1

Length = 2079 bits

Message:

b546430b f87b4f1e e834704c d6951c36 e26f108c f731788f 48dc34f1 678c0522 1c8fa7ff
2f39f477 e7e49ef6 0a4ec2c3 de24312a 96aa26e1 cfba5756 3838b297 f47e8510 c779fd66

```
54b14338 6fa639d3 1edbd6c0 6e47d159 d94362f2 6aeeedee 0e4f49d9 bf841299 5415bfad
56ee82d1 ca7463ab f085b082 b09904d6 d990d43c f2e062f4 0839d932 48b1eb92 cdfed530
0bc14828 0430b6d0 caa094b6 ec8911ab 7dc36824 b824dc0a f6682b09 35fde7b4 92a14dc2
f4364803 8da2cf79 170d2d50 133fd494 16cb6e33 bea90b8b f4559b03 732a01ea 290e6d07
4f79bb83 c10e5800 15cc1a85 b36b5501 046e9c4b dcae5135 690b8666 bd54b7a7 03ea7b6f
220a5469 a568027e
```

MAC: 109a1efe

5.6 Test Set 5

```
Key       = 6b 8b 08 ee 79 e0 b5 98 2d 6d 12 8e a9 f2 20 cb

Count     = 561eb2dd

Bearer    = 1c

Direction = 0

Length    = 5670 bits

Message:

    5bad7247 10ba1c56 d5a315f8 d40f6e09 3780be8e 8de07b69 92432018 e08ed96a 5734af8b
    ad8a575d 3a1f162f 85045cc7 70925571 d9f5b94e 454a77c1 6e72936b f016ae15 7499f054
    3b5d52ca a6dbeab6 97d2bb73 e41b8075 dce79b4b 86044f66 1d4485a5 43dd7860 6e0419e8
    059859d3 cb2b67ce 0977603f 81ff839e 33185954 4cfbc8d0 0fef1a4c 8510fb54 7d6b06c6
    11ef44f1 bce107cf a45a06aa b360152b 28dc1ebe 6f7fe09b 0516f9a5 b02a1bd8 4bb0181e
    2e89e19b d8125930 d178682f 3862dc51 b636f04e 720c47c3 ce51ad70 d94b9b22 55fbae90
    6549f499 f8c6d399 47ed5e5d f8e2def1 13253e7b 08d0a76b 6bfc68c8 12f375c7 9b8fe5fd
    85976aa6 d46b4a23 39d8ae51 47f680fb e70f978b 38effd7b 2f7866a2 2554e193 a94e98a6
    8b74bd25 bb2b3f5f b0a5fd59 887f9ab6 8159b717 8d5b7b67 7cb546bf 41eadca2 16fc1085
    0128f8bd ef5c8d89 f96afa4f a8b54885 565ed838 a950fee5 f1c3b0a4 f6fb71e5 4dfd169e
    82cecc72 66c850e6 7c5ef0ba 960f5214 060e71eb 172a75fc 1486835c bea65344 65b055c9
    6a72e410 52241823 25d83041 4b40214d aa8091d2 e0fb010a e15c6de9 0850973b df1e423b
    e148a237 b87a0c9f 34d4b476 05b803d7 43a86a90 399a4af3 96d3a120 0a62f3d9 507962e8
    e5bee6d3 da2bb3f7 237664ac 7a292823 900bc635 03b29e80 d63f6067 bf8e1716 ac25beba
    350deb62 a99fe031 85eb4f69 937ecd38 7941fda5 44ba67db 09117749 38b01827 bcc69c92
    b3f772a9 d2859ef0 03398b1f 6bbad7b5 74f7989a 1d10b2df 798e0dbf 30d65874 64d24878
    cd00c0ea ee8a1a0c c753a279 79e11b41 db1de3d5 038afaf4 9f5c682c 3748d8a3 a9ec54e6
    a371275f 1683510f 8e4f9093 8f9ab6e1 34c2cfdf 4841cba8 8e0cff2b 0bcc8e6a dcb71109
    b5198fec f1bb7e5c 531aca50 a56a8a3b 6de59862 d41fa113 d9cd9578 08f08571 d9a4bb79
    2af271f6 cc6dbb8d c7ec36e3 6be1ed30 8164c31c 7c0afc54 1c000000
```

MAC: e48a4e66

<End of Document>