ETSI/SAGE	**Version: 1.0**
Specification	**Date: 10th January 2006**

Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2&UIA2

Document 4: Design Conformance Test Data

Blank Page

PREFACE

This specification has been prepared by the 3GPP Task Force, and gives black-box test data for the algorithm set. 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 last of four, which between them form the entire specification of the 3GPP Confidentiality and Integrity Algorithms:

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***.
  Document 1: Algorithm Specifications.

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***.
  Document 2: SNOW 3G Algorithm Specification.

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***.
  Document 3: Implementors' Test Data.

- Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***.
  Document 4: Design Conformance Test Data.

This document is purely informative. The normative part of the specification of the***UEA2*** (confidentiality) and the ***UIA2*** (integrity) algorithms is in the main body of document 1. The normative part of the specification of **SNOW 3G** is found in document 2.

Blank Page

TABLE OF CONTENTS

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 ***UEA2*** & ***UIA2***. Document 1: ***UEA2*** and ***UIA2*** specifications.

[4] Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***. Document 2: **SNOW 3G** specification.

[5] Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***. Document 3: Implementors' Test Data.

[6] Specification of the 3GPP Confidentiality and Integrity Algorithms ***UEA2*** & ***UIA2***. Document 4: Design Conformance Test Data.

[7] P. Ekdahl and T. Johansson, "A new version of the stream cipher SNOW", in Selected Areas in Cryptology (SAC 2002), LNCS 2595, pp. 47–61, Springer-Verlag,

1. OUTLINE OF THE DESIGN CONFORMANCE TEST DATA

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

Section 3 provides test data for the confidentiality algorithm ***UEA2***.

Section 4 provides test data for the integrity algorithm ***UIA2***.

2. INTRODUCTORY INFORMATION

2.1. Introduction

Within the security architecture of the 3GPP system there are two standardised algorithms; a confidentiality algorithm ***UEA2***, and an integrity algorithm ***UIA2***. 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 ***UEA2*** and ***UIA2*** algorithms.

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 ***UEA2*** and ***UIA2*** 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).

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 ***UEA2*** is run every entry in the internal S-boxes S_R and S_Q will have been accessed.

3. CONFIDENTIALITY ALGORITHM *UEA2*

3.1. Overview

The test data sets presented here are for the ***UEA2*** confidentiality algorithm.

3.2. 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.

3.3. Test Set 1

3.3.1. Binary Representation

```
Key= 1101001111000101110101011001001000110010011111111011000100011100
     0100000000110101110001100110100000001010111110001100011011010001
Count     = 00111001100010100101100110110100
Bearer    = 10101
Direction = 1
Length    = 253 bits
Plaintext:
1001100000011011101001101000001001001100000110111111101100011010
1011010010000101010001110010000000101001101101110001110110000000
1000110011100011001111100010110011000011110000001011010111111100
00011111001111011110100010100110110111000110011010110001111100

Cipher text:
0101110101011011111111100111010111101011000001001111011010001100
1110000010100001001000110111011111101010000000001011001101111101
0100011111000110101000001011101000000110001100001001000101010101
00001000011010101000010110011100010000110100000110110011011110
```

3.3.2. Hexadecimal Representation

```
Key       = D3C5D592327FB11C4035C6680AF8C6D1
Count     = 398A59B4
Bearer    = 15
Direction = 1
Length    = 253 bits
Plaintext:
981BA682 4C1BFB1A B4854720 29B71D80 8CE33E2C C3C0B5FC 1F3DE8A6
DC66B1F0

Cipher text:
5D5BFE75 EB04F68C E0A12377 EA00B37D 47C6A0BA 06309155 086A859C
4341B378
```

3.4. Test Set 2

```
Key       = 2BD6459F82C440E0952C49104805FF48
Count     = C675A64B
Bearer    = 0C
Direction = 1
Length    = 798 bits
Plaintext:
7EC61272 743BF161 4726446A 6C38CED1 66F6CA76 EB543004 4286346C
EF130F92 922B0345 0D3A9975 E5BD2EA0 EB55AD8E 1B199E3E C4316020
E9A1B285 E7627953 59B7BDFD 39BEF4B2 484583D5 AFE082AE E638BF5F
D5A60619 3901A08F 4AB41AAB 9B134880

Cipher text:
3F678507 14B8DA69 EFB727ED 7A6C0C50 714AD736 C4F56000 06E3525B
E807C467 C677FF86 4AF45FBA 09C27CDE 38F87A1F 84D59AB2 55408F2C
7B82F9EA D41A1FE6 5EABEBFB C1F3A4C5 6C9A26FC F7B3D66D 0220EE47
75BC5817 0A2B12F3 431D11B3 44D6E36C
```

3.5. Test Set 3

```
Key       = 0A8B6BD8D9B08B08D64E32D1817777FB
Count     = 544D49CD
Bearer    = 04
Direction = 0
Length    = 310 bits
Plaintext:
FD40A41D 370A1F65 74509568 7D47BA1D 36D2349E 23F64439 2C8EA9C4
9D40C132 71AFF264 D0F248

Cipher text:
48148E54 52A210C0 5F46BC80 DC6F7349 5B02048C 1B958B02 6102CA97
280279A4 C18D2EE3 08921C
```

3.6. Test Set 4

```
Key       = AA1F95AEA533BCB32EB63BF52D8F831A
Count     = 72D8C671
Bearer    = 10
Direction = 1
Length    = 1022 bits
Plaintext:
```

```
FB1B96C5 C8BADFB2 E8E8EDFD E78E57F2 AD81E741 03FC430A 534DCC37
AFCEC70E 1517BB06 F27219DA E49022DD C47A068D E4C9496A 951A6B09
EDBDC864 C7ADBD74 0AC50C02 2F3082BA FD22D781 97C5D508 B977BCA1
3F32E652 E74BA728 576077CE 628C535E 87DC6077 BA07D290 68590C8C
B5F1088E 082CFA0E C961302D 69CF3D44

Cipher text:
FFCFC2FE AD6C094E 96C589D0 F6779B67 84246C3C 4D1CEA20 3DB3901F
40AD4FD7 138BC6D7 7E8320CB 102F497F DD44A269 A96ECB28 617700E3
32EB2F73 6B34F4F2 693094E2 2FF94F9B E4723DA4 0C40DFD3 931CC1AC
9723F6B4 A9913E96 B6DB7ABC ACE41517 7C1D0115 C5F09B5F DEA0B3AD
B8F9DA6E 9F9A04C5 43397B9D 43F87330
```

3.7. Test Set 5

```
Key       = 9618AE46891F86578EEBE90EF7A1202E
Count     = C675A64B
Bearer    = 0C
Direction = 1
Length    = 1245 bits
Plaintext:
8DAA17B1 AE050529 C6827F28 C0EF6A12 42E93F8B 314FB18A 77F790AE
049FEDD6 12267FEC AEFC4501 74D76D9F 9AA7755A 30CD90A9 A5874BF4
8EAF70EE A3A62A25 0A8B6BD8 D9B08B08 D64E32D1 817777FB 544D49CD
49720E21 9DBF8BBE D33904E1 FD40A41D 370A1F65 74509568 7D47BA1D
36D2349E 23F64439 2C8EA9C4 9D40C132 71AFF264 D0F24841 D6465F09
96FF84E6 5FC517C5 3EFC3363 C38492A8

Cipher text:
6CDB18A7 CA8218E8 6E4B4B71 6A4D0437 1FBEC262 FC5AD0B3 819B187B
97E55B1A 4D7C19EE 24C8B4D7 723CFEDF 045B8ACA E4869517 D80E5061
5D9035D5 D9C5A40A F602280B 542597B0 CB18619E EB359257 59D195E1
00E8E4AA 0C38A3C2 ABE0F3D8 FF04F3C3 3C295069 C23694B5 BBEACDD5
42E28E8A 94EDB911 9F412D05 4BE1FA72 72B5FFB2 B2570F4F 7CEAF383
A8A9D935 72F04D6E 3A6E2937 26EC62C8
```

3.8. Test Set 6

```
Key       = 54F4E2E04C83786EEC8FB5ABE8E36566
Count     = ACA4F50F
Bearer    = 0B
Direction = 0
Length    = 3861 bits
Plaintext:

40981BA6 824C1BFB 4286B299 783DAF44 2C099F7A B0F58D5C 8E46B104
F08F01B4 1AB48547 2029B71D 36BD1A3D 90DC3A41 B46D5167 2AC4C966
3A2BE063 DA4BC8D2 808CE33E 2CCCBFC6 34E1B259 060876A0 FBB5A437
EBCC8D31 C19E4454 318745E3 FA16BB11 ADAE2488 79FE52DB 2543E53C
F445D3D8 28CE0BF5 C560593D 97278A59 762DD0C2 C9CD68D4 496A7925
08614014 B13B6AA5 1128C18C D6A90B87 978C2FF1 CABE7D9F 898A411B
FDB84F68 F6727B14 99CDD30D F0443AB4 A6665333 0BCBA110 5E4CEC03
4C73E605 B4310EAA ADCFD5B0 CA27FFD8 9D144DF4 79275942 7C9CC1F8
CD8C8720 2364B8A6 87954CB0 5A8D4E2D 99E73DB1 60DEB180 AD0841E9
6741A5D5 9FE4189F 15420026 FE4CD121 04932FB3 8F735340 438AAF7E
CA6FD5CF D3A195CE 5ABE6527 2AF607AD A1BE65A6 B4C9C069 3234092C
4D018F17 56C6DB9D C8A6D80B 88813861 6B681262 F954D0E7 71174878
0D92291D 86299972 DB741CFA 4F37B8B5 6CDB18A7 CA8218E8 6E4B4B71
```

```
6A4D0437 1FBEC262 FC5AD0B3 819B187B 97E55B1A 4D7C19EE 24C8B4D7
723CFEDF 045B8ACA E4869517 D80E5061 5D9035D5 D9C5A40A F602280B
542597B0 CB18619E EB359257 59D195E1 00E8E4AA 0C38A3C2 ABE0F3D8
FF04F3C3 3C295069 C23694B5 BBEACDD5 42E28E8A 94EDB911 9F412D05
4BE1FA72 B09550
```

Cipher text:

```
351E30D4 D910C5DD 5AD7834C 426E6C0C AB6486DA 7B0FDA4C D83AF1B9
647137F1 AC43B434 223B19BE 07BD89D1 CC306944 D3361EA1 A2F8CDBD
32165597 6350D00B 80DD8381 20A7755C 6DEA2AB2 B0C99A91 3F47DAE2
B8DEB9A8 29E5469F F2E18777 6F6FD081 E3871D11 9A76E24C 917EA626
48E02E90 367564DE 72AE7E4F 0A4249A9 A5B0E465 A2D6D9DC 87843B1B
875CC9A3 BE93D8DA 8F56ECAF 5981FE93 C284318B 0DEC7A3B A108E2CB
1A61E966 FA7AFA7A C7F67F65 BC4A2DF0 70D4E434 845F109A B2B68ADE
3DC316CA 6332A628 93E0A7EC 0B4FC251 91BF2FF1 B9F9815E 4BA8A99C
643B5218 04F7D585 0DDE3952 206EC6CC F340F9B3 220B3023 BDD06395
6EA83339 20FDE99E 0675410E 49EF3B4D 3FB3DF51 92F99CA8 3D3B0032
DE08C220 776A5865 B0E4B3B0 C75DEFE7 762DFF01 8EA7F5BE 2B2F972B
2A8BA597 0E43BD6F DD63DAE6 29784EC4 8D610054 EE4E4B5D BBF1FC2F
A0B830E9 4DCBB701 4E8AB429 AB100FC4 8F83171D 99FC258B 7C2BA7C1
76EAEAAD 37F860D5 97A31CE7 9B594733 C7141DF7 9151FCA9 0C08478A
5C6C2CC4 81D51FFE CE3CD7D2 58134882 7A71F091 428EBE38 C95A3F5C
63E056DF B7CC45A9 B7C07D83 4E7B20B9 9ED20242 9C14BB85 FFA43B7C
B68495CD 75AB66D9 64D4CAFE 64DD9404 DAE2DC51 10617F19 4FC3C184
F583CD0D EF6D00
```

4. INTEGRITY ALORITHM *UIA2*

4.1. Overview

The test data sets presented here are for the ***UIA2*** 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]
Fresh = FRESH[0] || … || FRESH[31]
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.

4.2. Test Set 1

4.2.1. Binary representation

```
Key= 0010101111010110010001011001111110000010110001011011001100000000
     1001010100101100010010010001000001001000100000011111111101001000
Count     = 00111000101001101111000001010110
Fresh     = 10111000101011101111110110101001
Direction = 0
Length    = 88 bits
Message:
0011001100110010001101000110001001100011001110010011100001100001
001101110011010001111001

Output: 11101110010000011001111000001101
```

4.2.2. Hexadecimal representation

```
Key       = 2BD6459F82C5B300952C49104881FF48
Count     = 38A6F056
Fresh     = B8AEFDA9
Direction = 0
Length    = 88 bits
Message:
3332346263393861 373479

Output: EE419E0D
```

4.3. Test Set 2

```
Key       = 7E5E94431E11D73828D739CC6CED4573
Count     = 36AF6144
Fresh     = 9838F03A
Direction = 1
Length    = 254 bits
Message:
B3D3C9170A4E1632 F60F861013D22D84 B726B6A278D802D1 EEAF1321BA5929DC

Output: 92F2A453
```

4.4. Test Set 3

```
Key       = D3419BE821087ACD02123A9248033359
Count     = C7590EA9
Fresh     = 57D5DF7D
Direction = 0
Length    = 511 bits
Message:
BBB057038809496B CFF86D6FBC8CE5B1 35A06B166054F2D5 65BE8ACE75DC851E
0BCDD8F07141C495 872FB5D8C0C66A8B 6DA556663E4E4612 05D84580BEE5BC7E

Output: AD8C69F9
```

4.5. Test Set 4

```
Key       = 83FD23A244A74CF358DA3019F1722635
Count     = 36AF6144
Fresh     = 4F302AD2
Direction = 1
Length    = 768 bits
Message:
35C68716633C66FB 750C266865D53C11 EA05B1E9FA49C839 8D48E1EFA5909D39
47902837F5AE96D5 A05BC8D61CA8DBEF 1B13A4B4ABFE4FB1 006045B674BB5472
9304C382BE53A5AF 05556176F6EAA2EF 1D05E4B083181EE6 74CDA5A485F74D7A

Output: 7306D607
```

4.6. Test Set 5

```
Key       = 6832A65CFF4473621EBDD4BA26A921FE
Count     = 36AF6144
Fresh     = 9838F03A
Direction = 0
Length    = 383 bits
Message:
D3C5383962682071 7765667620323837 636240981BA6824C 1BFB1AB485472029
B71D808CE33E2CC3 C0B5FC1F3DE8A6DC

Output: E3D36EF1
```

4.7. Test Set 6

```
Key       = 5D0A80D8134AE19677824B671E838AF4
Count     = 7827FAB2
Fresh     = A56C6CA2
Direction = 1
Length    = 2558 bits
```

```
Message:
70DEDF2DC42C5CBD 3A96F8A0B11418B3 608D5733604A2CD3 6AABC70CE3193BB5
153BE2D3C06DFDB2 D16E9C357158BE6A 41D6B861E491DB3F BFEB518EFCF048D7
D58953730FF30C9E C470FFCD663DC342 01C36ADDC0111C35 B38AFEE7CFDB582E
3731F8B4BAA8D1A8 9C06E81199A97162 27BE344EFCB436DD D0F096C064C3B5E2
C399993FC77394F9 E09720A811850EF2 3B2EE05D9E617360 9D86E1C0C18EA51A
012A00BB413B9CB8 188A703CD6BAE31C C67B34B1B00019E6 A2B2A690F02671FE
7C9EF8DEC0094E53 3763478D58D2C5F5 B827A0148C5948A9 6931ACF84F465A64
E62CE74007E991E3 7EA823FA0FB21923 B79905B733B631E6 C7D6860A3831AC35
1A9C730C52FF72D9 D308EEDBAB21FDE1 43A0EA17E23EDC1F 74CBB3638A2033AA
A15464EAA733385D BBEB6FD73509B857 E6A419DCA1D8907A F977FBAC4DFA35EC

Output: C058D244
```

4.8. Test Set 7

```
Key       = B3120FFDB2CF6AF4E73EAF2EF4EBEC69
Count     = 296F393C
Fresh     = 6B227737
Direction = 1
Length    = 16448 bits

Message:
0000000000000000 0101010101010101 E0958045F3A0BBA4 E3968346F0A3B8A7
C02A018AE6407652 26B987C913E6CBF0 83570016CF83EFBC 61C082513E21561A
427C009D28C298EF ACE78ED6D56C2D45 05AD032E9C04DC60 E73A81696DA665C6
C48603A57B45AB33 221585E68EE31691 87FB0239528632DD 656C807EA3248B7B
46D002B2B5C7458E B85B9CE95879E034 0859055E3B0ABBC3 EACE8719CAA80265
C97205D5DC4BCC90 2FE1839629ED7132 8A0F0449F588557E 6898860E042AECD8
4B2404C212C9222D A5BF8A89EF679787 0CF50771A60F66A2 EE62853657ADDF04
CDDE07FA414E11F1 2B4D81B9B4E8AC53 8EA30666688D881F 6C348421992F31B9
4F8806ED8FCCFF4C 9123B89642527AD6 13B109BF75167485 F1268BF884B4CD23
D29A0934925703D6 34098F7767F1BE74 91E708A8BB949A38 73708AEF4A36239E
50CC08235CD5ED6B BE578668A17B58C1 171D0B90E813A9E4 F58A89D719B11042
D6360B1B0F52DEB7 30A58D58FAF46315 954B0A8726914759 77DC88C0D733FEFF
54600A0CC1D0300A AAEB94572C6E95B0 1AE90DE04F1DCE47 F87E8FA7BEBF77E1
DBC20D6BA85CB914 3D518B285DFA04B6 98BF0CF7819F20FA 7A288EB0703D995C
59940C7C66DE57A9 B70F82379B70E203 1E450FCFD2181326 FCD28D8823BAAA80
DF6E0F4435596475 39FD8907C0FFD9D7 9C130ED81C9AFD9B 7E848C9FED38443D
5D380E53FBDB8AC8 C3D3F06876054F12 2461107DE92FEA09 C6F6923A188D53AF
E54A10F60E6E9D5A 03D996B5FBC820F8 A637116A27AD04B4 44A0932DD60FBD12
671C11E1C0EC73E7 89879FAA3D42C64D 20CD1252742A3768 C25A901585888ECE
E1E612D9936B403B 0775949A66CDFD99 A29B1345BAA8D9D5 400C91024B0A6073
63B013CE5DE9AE86 9D3B8D95B0570B3C 2D391422D32450CB CFAE96652286E96D
EC1214A934652798 0A8192EAC1C39A3A AF6F15351DA6BE76 4DF89772EC0407D0
6E4415BEFAE7C925 80DF9BF507497C8F 2995160D4E218DAA CB02944ABF83340C
E8BE1686A960FAF9 0E2D90C55CC6475B ABC3171A80A36317 4954955D7101DAB1
6AE8179167E21444 B443A9EAAA7C91DE 36D118C39D389F8D D4469A846C9A262B
F7FA18487A79E8DE 11699E0B8FDF557C B48719D453BA7130 56109B93A218C896
75AC195FB4FB0663 9B3797144955B3C9 327D1AEC003D42EC D0EA98ABF19FFB4A
F3561A67E77C35BF 15C59C2412DA881D B02B1BFBCEBFAC51 52BC99BC3F1D15F7
71001B7029FEDB02 8F8B852BC4407EB8 3F891C9CA733254F DD1E9EDB56919CE9
FEA21C174072521C 18319A54B5D4EFBE BDDF1D8B69B1CBF2 5F489FCC98137254
7CF41D008EF0BCA1 926F934B735E090B 3B251EB33A36F82E D9B29CF4CB944188
FA0E1E38DD778F7D 1C9D987B28D132DF B9731FA4F4B41693 5BE49DE30516AF35
78581F2F13F561C0 663361941EAB249A 4BC123F8D15CD711 A956A1BF20FE6EB7
8AEA2373361DA042 6C79A530C3BB1DE0 C99722EF1FDE39AC 2B00A0A8EE7C800A
08BC2264F89F4EFF E627AC2F0531FB55 4F6D21D74C590A70 ADFAA390BDFBB3D6
8E46215CAB187D23 68D5A71F5EBEC081 CD3B20C082DBE4CD 2FACA28773795D6B
```

0C10204B659A939E F29BBE1088243624 429927A7EB576DD3 A00EA5E01AF5D475
83B2272C0C161A80 6521A16FF9B0A722 C0CF26B025D5836E 2258A4F7D4773AC8
01E4263BC294F43D EF7FA8703F3A4197 46352588765 2B0B2 A4A2A7CF87F00914
871E25039113C7E1 618DA34064B57A43 C463249FB8D05E0F 26F4A6D84972E7A9
054824145F91295C DBE39A6F920FACC6 59712B46A54BA295 BBE6A90154E91B33
985A2BCD420AD5C6 7EC9AD8EB7AC6864 DB272A516BC94C28 39B0A8169A6BF58E
1A0C2ADA8C883B7B F497A49171268ED1 5DDD2969384E7FF4 BF4AAB2EC9ECC652
9CF629E2DF0F08A7 7A65AFA12AA9B505 DF8B287EF6CC9149 3D1CAA39076E28EF
1EA028F5118DE61A E02BB6AEFC3343A0 50292F199F401857 B2BEAD5E6EE2A1F1
91022F9278016F04 7791A9D18DA7D2A6 D27F2E0E51C2F6EA 30E8AC49A0604F4C
13542E85B68381B9 FDCFA0CE4B2D3413 54852D360245C536 B612AF71F3E77C90
95AE2DBDE504B265 733DABFE10A20FC7 D6D32C21CCC72B8B 3444AE663D65922D
17F82CAA2B865CD8 8913D291A6589902 6EA1328439723C19 8C36B0C3C8D085BF
AF8A320FDE334B4A 4919B44C2B95F6E8 ECF73393F7F0D2A4 0E60B1D406526B02
2DDC331810B1A5F7 C347BD53ED1F105D 6A0D30ABA477E178 889AB2EC55D558DE
AB2630204336962B 4DB5B663B6902B89 E85B31BC6AF50FC5 0ACCB3FB9B57B663
297031378DB47896 D7FBAF6C600ADD2C 67F936DB037986DB 856EB49CF2DB3F7D
A6D23650E438F188 4041B013119E4C2A E5AF37CCCDFB6866 0738B58B3C59D1C0
248437472ABA1F35 CA1FB90CD714AA9F 635534F49E7C5BBA 81C2B6B36FDEE21C
A27E347F793D2CE9 44EDB23C8C9B914B E10335E350FEB507 0394B7A4A15C0CA1
20283568B7BFC254 FE838B137A2147CE 7C113A3A4D65499D 9E86B87DBCC7F03B
BD3A3AB1AA243ECE 5BA9BCF25F82836C FE473B2D83E7A720 1CD0B96A72451E86
3F6C3BA664A6D073 D1F7B5ED990865D9 78BD3815D06094FC 9A2ABA5221C22D5A
B996389E3721E3AF 5F05BEDDC2875E0D FAEB39021EE27A41 187CBB45EF40C3E7
3BC03989F9A30D12 C54BA7D2141DA8A8 75493E65776EF35F 97DEBC2286CC4AF9
B4623EEE902F840C 52F1B8AD658939AE F71F3F72B9EC1DE2 1588BD35484EA444
36343FF95EAD6AB1 D8AFB1B2A303DF1B 71E53C4AEA6B2E3E 9372BE0D1BC99798
B0CE3CC10D2A596D 565DBA82F88CE4CF F3B33D5D24E9C083 1124BF1AD54B7925
32983DD6C3A8B7D0

Output: 179F2FA6

<End of Document>