INTERNATIONAL STANDARD

ISO/IEC 18033-4

Second edition
2011-12-15

AMENDMENT 1
2020-08

Information technology — Security techniques — Encryption algorithms —

Part 4: Stream ciphers

AMENDMENT 1: ZUC

Technologies de l'information — Techniques de sécurité — Algorithmes de chiffrement —

Partie 4: Chiffrements en flot

AMENDEMENT 1: ZUC

Reference number
ISO/IEC 18033-4:2011/Amd.1:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection.*

A list of all parts in the ISO/IEC 18033 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Information technology — Security techniques — Encryption algorithms —

Part 4: Stream ciphers

AMENDMENT 1: ZUC

Introduction

Change the last paragraph as follows:

This document includes six dedicated keystream generators:

— MUGI keystream generator;

— SNOW 2.0 keystream generator;

— Rabbit keystream generator;

— Decim^{v2} keystream generator;

— KCipher-2 (K2) keystream generator; and

— ZUC keystream generator.

4.1

Add the following symbols:

L_1 Linear transform with index 1 used for ZUC.

L_2 Linear transform with index 2 used for ZUC.

SS Subfunction used for ZUC.

*SUB*1 Lookup table with index 1 used for ZUC.

*SUB*2 Lookup table with index 2 used for ZUC.

8.6

Add new subclause 8.6 as follows:

8.6 ZUC keystream generator

8.6.1 Introduction to ZUC

ZUC is a keystream generator which uses as input a 128-bit secret key K and a 128-bit initialization vector IV. These are used to initialize state variables S_i ($i \geq 0$). The bit/byte order is big-endian, i.e., if the key and initialization vector are given as a sequence of bits/bytes, the first/leftmost bit/byte is the

most significant bit/byte of the corresponding data. It outputs a 32-bit keystream Z_i at every iteration of the function *Strm*.

The state variable S_i consists of two components. The first consists of sixteen 31-bit variables:

$$A^{(i)} = (A_{15}{}^{(i)}, A_{14}{}^{(i)}, \ldots., A_0{}^{(i)}),$$

and maintains the state of a linear feedback shift register. The second consists of two 32-bit variables:

$$R^{(i)} = (R_2{}^{(i)}, R_1{}^{(i)}),$$

that maintains the state of a finite state machine. ZUC is summarised in Figure 15, which shows a snapshot if its operation, at time *i*, omitting the time-dependent variable (*i*) from the notation.

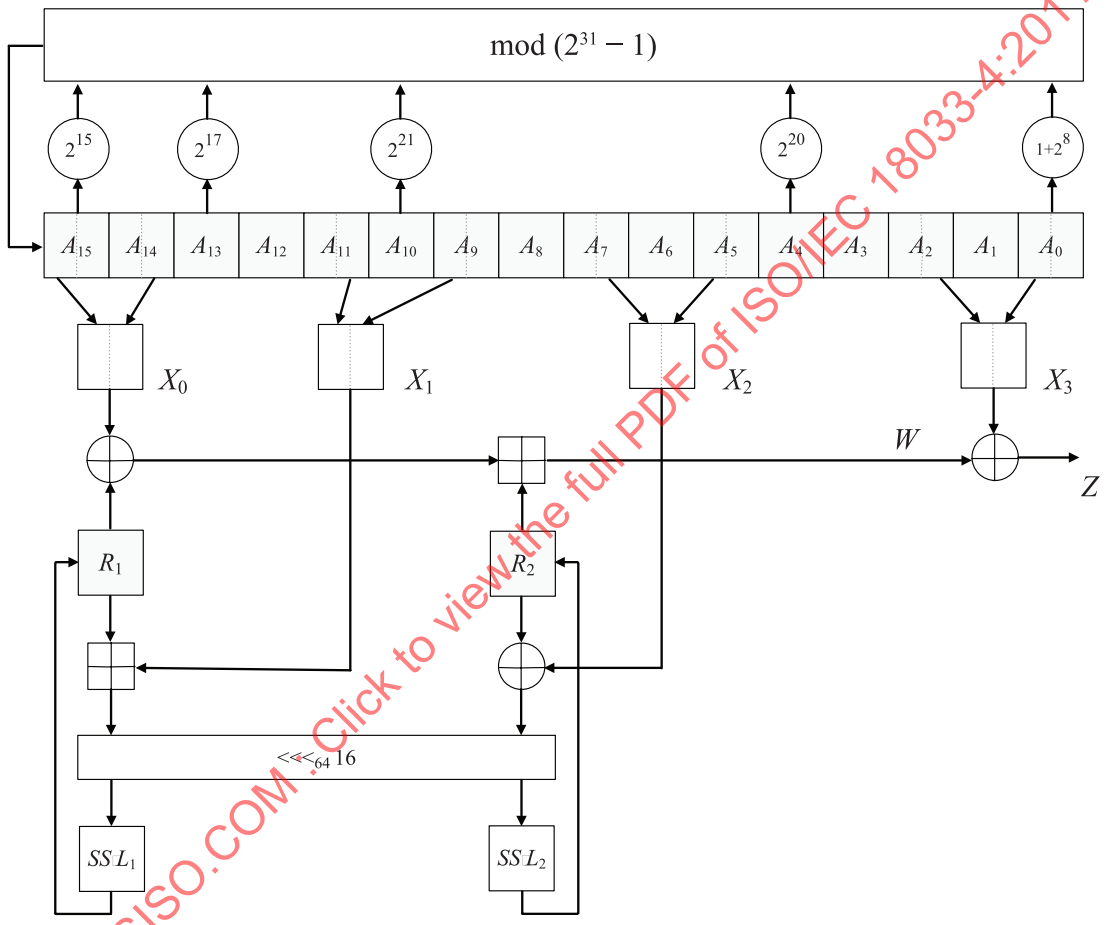

Figure 15 — Schematic drawing of ZUC

The *Init* function, defined in detail in 8.6.2, takes as input the 128-bit key *K* and the 128-bit initialization vector *IV*, and produces the initial value of the state variable $S_0 = (A^{(0)}, R^{(0)})$.

The *Next* function, defined in detail in 8.6.3, takes as input the state variable $S_i = (A^{(i)}, R^{(i)})$ and produces as output the next value of the state variable $S_{i+1} = (A^{(i+1)}, R^{(i+1)})$. The *Next* function runs in two modes, depending on whether the iteration performed is part of the initialization mode or of the normal mode of generating output.

The *Strm* function, defined in detail in 8.6.4, takes as input the state variable $S_i = (A^{(i)}, R^{(i)})$ and produces as output the 32-bit keystream Z_i.

NOTE See document [20] for theoretical background on the design rationale for ZUC.

A 240-bit constant $D = d_0 \| d_1 \| \ldots \| d_{15}$ used in ZUC:

d_0 = 100010011010111, d_1 = 010011010111100, d_2 = 110001001101011, d_3 = 001001101011110,

d_4 = 101011110001001, d_5 = 011010111100010, d_6 = 111000100110101, d_7 = 000100110101111,

d_8 = 100110101111000, d_9 = 010111100010011, d_{10} = 110101111000100, d_{11} = 001101011110001,

d_{12} = 101111000100110, d_{13} = 011110001001101, d_{14} = 111100010011010, d_{15} = 100011110101100,

where for i = 0, 1, ..., 15, d_i is a 15-bit variable in binary notation.

The description uses notations defined in Clause 4 of this part of ISO/IEC 18033. For a string A which has at least 16 bits, the notation A_H represents the leftmost 16 bits of A and the notation A_L represents the rightmost 16 bits of A. For example, if A = 10001001101111101111101011110011 is a 31-bit string, then A_H =1000100110111110 and A_L = 0111110101111001.

8.6.2 Initialization function *Init*

The Initialization function *Init* is as follows.

Input: 128-bit key K, 128-bit initialization vector IV.

Output: Initial value of state variable $S_0 = (A^{(0)}, R^{(0)})$.

a) Initialize the state variable S_{-33} with the key K, the 128-bit initialization vector IV and the constant D.

- Set $(k_0, k_1, \ldots, k_{15}) = K$; $(iv_0, iv_1, \ldots, iv_{15}) = IV$, where k_i and iv_i are bytes for i = 0, 1, ..., 15.
- Set $A_i^{(-33)} = k_i \| d_i \| iv_i$ for i = 0, 1, ..., 15.
- Set $R_1^{(-33)} = R_2^{(-33)} = 0^{(32)}$.

b) Set $S_{-1} = Next^{32}(S_{-33}, \text{INIT})$, where $Next^{32}$ denotes 32 iterations of the *Next* function.

c) Set $S_0 = Next(S_{-1}, \text{null})$.

d) Output S_0.

8.6.3 Next-state function *Next*

The *Next* function has two modes, and is defined as follows.

Input: State variable $S_i = (A^{(i)}, R^{(i)})$, mode ∈ {INIT, null}.

Output: Next value of the state variable $S_{i+1} = (A^{(i+1)}, R^{(i+1)})$.

Local variables: 32-bit strings W, W_1, W_2, X_0, X_1, X_2 and 31-bit string V.

a) Set $X_0 = A_{15}{}^{(i)}{}_H \| A_{14}{}^{(i)}{}_L$; $X_1 = A_{11}{}^{(i)}{}_L \| A_9{}^{(i)}{}_H$; $X_2 = A_7{}^{(i)}{}_L \| A_5{}^{(i)}{}_H$.

b) Set $W = (X_0 \oplus R_1^{(i)}) +_{32} R_2^{(i)}$; $W_1 = R_1^{(i)} +_{32} X_1$; $W_2 = R_2^{(i)} \oplus X_2$; $R_1^{(i+1)} = SS(L_1(W_{1L} \| W_{2H}))$; $R_2^{(i+1)} = SS(L_2(W_{2L} \| W_{1H}))$.

c) Set $V = 2^{15}A_{15}^{(i)} + 2^{17}A_{13}^{(i)} + 2^{21}A_{10}^{(i)} + 2^{20}A_4^{(i)} + (1+2^8)A_0^{(i)} \bmod (2^{31}-1)$.

d) If mode = INIT, set $A_{15}^{(i+1)} = V + (31 \sim W) \bmod (2^{31}-1)$. Otherwise, set $A_{15}^{(i+1)} = V$. If $A_{15}^{(i+1)} = 0$, set $A_{15}^{(i+1)} = 2^{31}-1$.

e) Set $A_j^{(i+1)} = A_{j+1}^{(i)}$ for j = 0, 1, ..., 14.

f) Set $S_{i+1} = (A^{(i+1)}, R^{(i+1)})$.

g) Output S_{i+1}.

NOTE For two 31-bit stings a and b, if $b = 2^i$, then $ab \bmod (2^{31}-1) = a <<<_{31} i \bmod (2^{31} - 1)$; if $b = 2^i + 2^j$, then $ab \bmod (2^{31}-1) = (a <<<_{31} i) + (a <<<_{31} j) \bmod (2^{31} - 1)$. Reference C code for ZUC is given in document [21].

8.6.4 Keystream function *Strm*

The keystream function *Strm* is as follows:

Input: State variable S_i.

Output: 32-bit keystream Z_i.

Local variables: 32-bit strings X_0, X_3.

a) Set $X_0 = A_{15}{}^{(i)}{}_{\mathrm{H}} \,||\, A_{14}{}^{(i)}{}_{\mathrm{L}}$; $X_3 = A_2{}^{(i)}{}_{\mathrm{L}} \,||\, A_0{}^{(i)}{}_{\mathrm{H}}$.

b) Set $Z_i = ((X_0 \oplus R_1{}^{(i)}) +_{32} R_2{}^{(i)}) \oplus X_3$.

c) Output Z_i.

8.6.5 Function *SS*

The function *SS* is as follows:

Input: 32-bit string X.

Output: 32-bit string Y.

— Define $X = x_3 \,||\, x_2 \,||\, x_1 \,||\, x_0$, where x_i is a byte for $i = 0, 1, 2, 3$.

— Set $Y = SUB1[x_3] \,||\, SUB2[x_2] \,||\, SUB1[x_1] \,||\, SUB2[x_0]$.

— Output Y.

The functions *SUB*1 and *SUB*2 are defined by the following substitution tables:

```
SUB1 [256] = {
0x3e,0x72,0x5b,0x47,0xca,0xe0,0x00,0x33,0x04,0xd1,0x54,0x98,0x09,0xb9,0x6d,0xcb,
0x7b,0x1b,0xf9,0x32,0xaf,0x9d,0x6a,0xa5,0xb8,0x2d,0xfc,0x1d,0x08,0x53,0x03,0x90,
0x4d,0x4e,0x84,0x99,0xe4,0xce,0xd9,0x91,0xdd,0xb6,0x85,0x48,0x8b,0x29,0x6e,0xac,
0xcd,0xc1,0xf8,0x1e,0x73,0x43,0x69,0xc6,0xb5,0xbd,0xfd,0x39,0x63,0x20,0xd4,0x38,
0x76,0x7d,0xb2,0xa7,0xcf,0xed,0x57,0xc5,0xf3,0x2c,0xbb,0x14,0x21,0x06,0x55,0x9b,
0xe3,0xef,0x5e,0x31,0x4f,0x7f,0x5a,0xa4,0x0d,0x82,0x51,0x49,0x5f,0xba,0x58,0x1c,
0x4a,0x16,0xd5,0x17,0xa8,0x92,0x24,0x1f,0x8c,0xff,0xd8,0xae,0x2e,0x01,0xd3,0xad,
0x3b,0x4b,0xda,0x46,0xeb,0xc9,0xde,0x9a,0x8f,0x87,0xd7,0x3a,0x80,0x6f,0x2f,0xc8,
0xb1,0xb4,0x37,0xf7,0x0a,0x22,0x13,0x28,0x7c,0xcc,0x3c,0x89,0xc7,0xc3,0x96,0x56,
0x07,0xbf,0x7e,0xf0,0x0b,0x2b,0x97,0x52,0x35,0x41,0x79,0x61,0xa6,0x4c,0x10,0xfe,
0xbc,0x26,0x95,0x88,0x8a,0xb0,0xa3,0xfb,0xc0,0x18,0x94,0xf2,0xe1,0xe5,0xe9,0x5d,
0xd0,0xdc,0x11,0x66,0x64,0x5c,0xec,0x59,0x42,0x75,0x12,0xf5,0x74,0x9c,0xaa,0x23,
0x0e,0x86,0xab,0xbe,0x2a,0x02,0xe7,0x67,0xe6,0x44,0xa2,0x6c,0xc2,0x93,0x9f,0xf1,
0xf6,0xfa,0x36,0xd2,0x50,0x68,0x9e,0x62,0x71,0x15,0x3d,0xd6,0x40,0xc4,0xe2,0x0f,
0x8e,0x83,0x77,0x6b,0x25,0x05,0x3f,0x0c,0x30,0xea,0x70,0xb7,0xa1,0xe8,0xa9,0x65,
0x8d,0x27,0x1a,0xdb,0x81,0xb3,0xa0,0xf4,0x45,0x7a,0x19,0xdf,0xee,0x78,0x34,0x60};

SUB2 [256] = {
0x55,0xc2,0x63,0x71,0x3b,0xc8,0x47,0x86,0x9f,0x3c,0xda,0x5b,0x29,0xaa,0xfd,0x77,
0x8c,0xc5,0x94,0x0c,0xa6,0x1a,0x13,0x00,0xe3,0xa8,0x16,0x72,0x40,0xf9,0xf8,0x42,
0x44,0x26,0x68,0x96,0x81,0xd9,0x45,0x3e,0x10,0x76,0xc6,0xa7,0x8b,0x39,0x43,0xe1,
0x3a,0xb5,0x56,0x2a,0xc0,0x6d,0xb3,0x05,0x22,0x66,0xbf,0xdc,0x0b,0xfa,0x62,0x48,
0xdd,0x20,0x11,0x06,0x36,0xc9,0xc1,0xcf,0xf6,0x27,0x52,0xbb,0x69,0xf5,0xd4,0x87,
0x7f,0x84,0x4c,0xd2,0x9c,0x57,0xa4,0xbc,0x4f,0x9a,0xdf,0xfe,0xd6,0x8d,0x7a,0xeb,
0x2b,0x53,0xd8,0x5c,0xa1,0x14,0x17,0xfb,0x23,0xd5,0x7d,0x30,0x67,0x73,0x08,0x09,
0xee,0xb7,0x70,0x3f,0x61,0xb2,0x19,0x8e,0x4e,0xe5,0x4b,0x93,0x8f,0x5d,0xdb,0xa9,
0xad,0xf1,0xae,0x2e,0xcb,0x0d,0xfc,0xf4,0x2d,0x46,0x6e,0x1d,0x97,0xe8,0xd1,0xe9,
0x4d,0x37,0xa5,0x75,0x5e,0x83,0x9e,0xab,0x82,0x9d,0xb9,0x1c,0xe0,0xcd,0x49,0x89,
0x01,0xb6,0xbd,0x58,0x24,0xa2,0x5f,0x38,0x78,0x99,0x15,0x90,0x50,0xb8,0x95,0xe4,
0xd0,0x91,0xc7,0xce,0xed,0x0f,0xb4,0x6f,0xa0,0xcc,0xf0,0x02,0x4a,0x79,0xc3,0xde,
0xa3,0xef,0xea,0x51,0xe6,0x6b,0x18,0xec,0x1b,0x2c,0x80,0xf7,0x74,0xe7,0xff,0x21,
0x5a,0x6a,0x54,0x1e,0x41,0x31,0x92,0x35,0xc4,0x33,0x07,0x0a,0xba,0x7e,0x0e,0x34,
```



```
0x88,0xb1,0x98,0x7c,0xf3,0x3d,0x60,0x6c,0x7b,0xca,0xd3,0x1f,0x32,0x65,0x04,0x28,
0x64,0xbe,0x85,0x9b,0x2f,0x59,0x8a,0xd7,0xb0,0x25,0xac,0xaf,0x12,0x03,0xe2,0xf2}.
```

8.6.6 Linear transforms L_1 and L_2

Both L_1 and L_2 are linear transforms of 32-bit strings, defined as follows:

$$L_1(X) = X \oplus (X <<<_{32} 2) \oplus (X <<<_{32} 10) \oplus (X <<<_{32} 18) \oplus (X <<<_{32} 24),$$

$$L_2(X) = X \oplus (X <<<_{32} 8) \oplus (X <<<_{32} 14) \oplus (X <<<_{32} 22) \oplus (X <<<_{32} 30).$$

Annex A

Replace the object identifiers in Annex A as follows:

```
EncryptionAlgorithms-4 {
    iso(1) standard(0) encryption-algorithms(18033) part(4)
        asn1-module(0) algorithm-object-identifiers(0) }
    DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- EXPORTS All; --

-- IMPORTS None; --

OID ::= OBJECT IDENTIFIER -- Alias

-- Synonyms --

is18033-4 OID ::= { iso(1) standard(0) is18033(18033) part4(4) }

id-kg OID ::= { is18033-4 keystream-generator(1) }
id-scmode OID ::= { is18033-4 stream-cipher-mode(2) }

-- Assignments --

id-kg-mugi OID ::= { id-kg mugi(1) }
id-kg-snow OID ::= { id-kg snow(2) }
id-kg-rabbit OID ::= { id-kg rabbit(3) }
id-kg-decim2 OID ::= { id-kg decim2(4) }
id-kg-k2 OID ::= { id-kg k2(5) }
id-kg-zuc OID ::= { id-kg zuc(6) }

id-scmode-additive OID ::= { id-scmode additive(1) }
id-scmode-multis01 OID ::= { id-scmode multis01(2) }

-- Algorithms and parameters --

StreamCipher ::= AlgorithmIdentifier {{ StreamCipherAlgorithms }}

StreamCipherAlgorithms ALGORITHM ::= {
  additiveStreamCipher |
  multiS01StreamCipher,

  -- Expect additional algorithms --
}

additiveStreamCipher ALGORITHM ::= {
    OID id-scmode-additive PARMS AdditiveStreamCipherParameters
}

AdditiveStreamCipherParameters ::= KeyGenerator

multiS01StreamCipher ALGORITHM ::= {
    OID id-scmode-multis01 PARMS MultiS01StreamCipherParameters
}

MultiS01StreamCipherParameters ::= SEQUENCE {
    keyGenerator KeyGenerator,
```

```
    securityParameter INTEGER DEFAULT 64,
    irreduciblePolynoial BIT STRING,
    redandancy BIT STRING,
    publicParameterR BIT STRING
        -- length determined by securityParameter
        -- for full interoperability multis01 parameters should
        -- include the padding method but they do not have object
        -- identifiers. for the time being they will have to be
        -- negotiated in an application-dependent way
}

KeyGenerator ALGORITHM ::= {
    mugiKeyGenerator |
    snowKeyGenerator |
    rabbitKeyGenerator |
    decim2KeyGenerator |
    k2KeyGenerator |
    zucKeyGenerator,

    ... -- Expect additional algorithms --
}

mugiKeyGenerator ALGORITHM ::= {
    OID id-kg-mugi PARMS NullParameters
}

snowKeyGenerator ALGORITHM ::= {
    OID id-kg-snow PARMS NullParameters
}

rabbitKeyGenerator ALGORITHM ::={
    OID id-kg-rabbit PARMS NullParameters
}

decim2KeyGenerator ALGORITHM ::={
    OID id-kg-decim2 PARMS NullParameters
}

k2KeyGenerator ALGORITHM ::={
    OID id-kg-k2 PARMS NullParameters
}

zucKeyGenerator ALGORITHM ::={
    OID id-kg-zuc PARMS NullParameters
}

NullParameters ::= NULL

-- Cryptographic algorithm identification --

ALGORITHM ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &Type OPTIONAL
}
    WITH SYNTAX { OID &id [PARMS &Type] }

AlgorithmIdentifier { ALGORITHM:IOSet } ::= SEQUENCE {
    algorithm ALGORITHM.&id( {IOSet} ),
    parameters ALGORITHM.&Type( {IOSet}{@algorithm} ) OPTIONAL
}

END -- EncryptionAlgorithms-4 --
```

C.7

Add new Clause C.7 as follows:

C.7 Example for ZUC

C.7.1 Key, initialization vector, and keystream triplets

```
k = 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
z = 27 be de 74 01 80 82 da 87 d4 e5 b6 9f 18 bf 66 32 07 0e 0f 39 b7 b6 92 b4 67 3e dc 31
84 a4 8e

k = 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
z = 06 57 cf a0 70 96 39 8b 73 4b 6c b4 88 3e ed f4 25 7a 76 eb 97 59 52 08 d8 84 ad cd b1
cb ff b8

k = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
iv= ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
z = 58 fb 51 5e 39 08 74 6d 7a 91 f2 34 49 4e d8 c8 51 2d 61 eb 69 3c 14 b8 cd 2d 3b fe 69
4f e8 1d

k = 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
z = 14 f1 c2 72 32 79 c4 19 4b 8e a4 1d 0c c8 08 63 d2 80 62 e1 e7 1d 3d da e3 c4 d1 58 a7
f0 67 ac
```

C.7.2 Sample internal states

```
k = 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
z = 14 f1 c2 72 32 79 c4 19 4b 8e a4 1d 0c c8 08 63 d2 80 62 e1 e7 1d 3d da e3 c4 d1 58 a7
f0 67 ac

Internal state at time -33
15:2dc7ac66 14:22f89ac7 13:3dbc4dd8 12:58de26fb 11:0e9af16b 10:326bc4da 09:47af136b
08:5acd781f
07:5709afca 06:7ef13515 05:4135e269 04:355789de 03:74935ea8 02:25e26b9a 01:2626bc31
00:1ec4d784
R1:00000000 R2:00000000

Internal state at time -32
15:3c7b93c0 14:2dc7ac66 13:22f89ac7 12:3dbc4dd8 11:58de26fb 10:0e9af16b 09:326bc4da
08:47af136b
07:5acd781f 06:5709afca 05:7ef13515 04:4135e269 03:355789de 02:74935ea8 01:25e26b9a
00:2626bc31
R1:9c62829f R2:5df00831

Internal state at time -31
15:41901ee9 14:3c7b93c0 13:2dc7ac66 12:22f89ac7 11:3dbc4dd8 10:58de26fb 09:0e9af16b
08:326bc4da
07:47af136b 06:5acd781f 05:5709afca 04:7ef13515 03:4135e269 02:355789de 01:74935ea8
00:25e26b9a
R1:3d533f3a R2:80ff1faf

Internal state at time -30
15:411efa99 14:41901ee9 13:3c7b93c0 12:2dc7ac66 11:22f89ac7 10:3dbc4dd8 09:58de26fb
08:0e9af16b
07:326bc4da 06:47af136b 05:5acd781f 04:5709afca 03:7ef13515 02:4135e269 01:355789de
00:74935ea8
R1:2ca57e9d R2:d1db72f9

Internal state at time -29
15:24b3f49f 14:411efa99 13:41901ee9 12:3c7b93c0 11:2dc7ac66 10:22f89ac7 09:3dbc4dd8
08:58de26fb
07:0e9af16b 06:326bc4da 05:47af136b 04:5acd781f 03:5709afca 02:7ef13515 01:4135e269
00:355789de
R1:0e8dc40f R2:60921a4f
```

```
Internal state at time -28
15:74265785 14:24b3f49f 13:411efa99 12:41901ee9 11:3c7b93c0 10:2dc7ac66 09:22f89ac7
08:3dbc4dd8
07:58de26fb 06:0e9af16b 05:326bc4da 04:47af136b 03:5acd781f 02:5709afca 01:7ef13515
00:4135e269
R1:16c81467 R2:da8e7d8a

Internal state at time -27
15:481c5b9d 14:74265785 13:24b3f49f 12:411efa99 11:41901ee9 10:3c7b93c0 09:2dc7ac66
08:22f89ac7
07:3dbc4dd8 06:58de26fb 05:0e9af16b 04:326bc4da 03:47af136b 02:5acd781f 01:5709afca
00:7ef13515
R1:50c9eaa4 R2:3c3b2dfd

Internal state at time -26
15:4b7f87ed 14:481c5b9d 13:74265785 12:24b3f49f 11:411efa99 10:41901ee9 09:3c7b93c0
08:2dc7ac66
07:22f89ac7 06:3dbc4dd8 05:58de26fb 04:0e9af16b 03:326bc4da 02:47af136b 01:5acd781f
00:5709afca
R1:59857b80 R2:be0fbdc1

Internal state at time -25
15:0e633ce7 14:4b7f87ed 13:481c5b9d 12:74265785 11:24b3f49f 10:411efa99 09:41901ee9
08:3c7b93c0
07:2dc7ac66 06:22f89ac7 05:3dbc4dd8 04:58de26fb 03:0e9af16b 02:326bc4da 01:47af136b
00:5acd781f
R1:9528f8ea R2:bcc7f7eb

Internal state at time -24
15:643ae5a6 14:0e633ce7 13:4b7f87ed 12:481c5b9d 11:74265785 10:24b3f49f 09:411efa99
08:41901ee9
07:3c7b93c0 06:2dc7ac66 05:22f89ac7 04:3dbc4dd8 03:58de26fb 02:0e9af16b 01:326bc4da
00:47af136b
R1:c59d2932 R2:e1098a64

Internal state at time -23
15:625ac5d7 14:643ae5a6 13:0e633ce7 12:4b7f87ed 11:481c5b9d 10:74265785 09:24b3f49f
08:411efa99
07:41901ee9 06:3c7b93c0 05:2dc7ac66 04:22f89ac7 03:3dbc4dd8 02:58de26fb 01:0e9af16b
00:326bc4da
R1:755ebae8 R2:3f9e6e86

Internal state at time -22
15:10e10abb 14:625ac5d7 13:643ae5a6 12:0e633ce7 11:4b7f87ed 10:481c5b9d 09:74265785
08:24b3f49f
07:411efa99 06:41901ee9 05:3c7b93c0 04:2dc7ac66 03:22f89ac7 02:3dbc4dd8 01:58de26fb
00:0e9af16b
R1:d643d938 R2:d799a5a3

Internal state at time -21
15:1bdc9fab 14:10e10abb 13:625ac5d7 12:643ae5a6 11:0e633ce7 10:4b7f87ed 09:481c5b9d
08:74265785
07:24b3f49f 06:411efa99 05:41901ee9 04:3c7b93c0 03:2dc7ac66 02:22f89ac7 01:3dbc4dd8
00:58de26fb
R1:1798b822 R2:92245168

Internal state at time -20
15:256b94a 14:1bdc9fab 13:10e10abb 12:625ac5d7 11:643ae5a6 10:0e633ce7 09:4b7f87ed
08:481c5b9d
07:74265785 06:24b3f49f 05:411efa99 04:41901ee9 03:3c7b93c0 02:2dc7ac66 01:22f89ac7
00:3dbc4dd8
R1:4e29d84e R2:61b91f59

Internal state at time -19
15:2af10db2 14:2567b94a 13:1bdc9fab 12:10e10abb 11:625ac5d7 10:643ae5a6 09:0e633ce7
08:4b7f87ed
07:481c5b9d 06:74265785 05:24b3f49f 04:411efa99 03:41901ee9 02:3c7b93c0 01:2dc7ac66
00:22f89ac7
R1:5b486570 R2:d97ebf32

Internal state at time -18
```

```
15:3448fcc0 14:2af10db2 13:2567b94a 12:1bdc9fab 11:10e10abb 10:625ac5d7 09:643ae5a6
08:0e633ce7
07:4b7f87ed 06:481c5b9d 05:74265785 04:24b3f49f 03:411efa99 02:41901ee9 01:3c7b93c0
00:2dc7ac66
R1:421fbdfa R2:effe033f

Internal state at time -17
15:789c639c 14:3448fcc0 13:2af10db2 12:2567b94a 11:1bdc9fab 10:10e10abb 09:625ac5d7
08:643ae5a6
07:0e633ce7 06:4b7f87ed 05:481c5b9d 04:74265785 03:24b3f49f 02:411efa99 01:41901ee9
00:3c7b93c0
R1:bff08d37 R2:28e1d53c

Internal state at time -16
15:10da5941 14:789c639c 13:3448fcc0 12:2af10db2 11:2567b94a 10:1bdc9fab 09:10e10abb
08:625ac5d7
07:643ae5a6 06:0e633ce7 05:4b7f87ed 04:481c5b9d 03:74265785 02:24b3f49f 01:411efa99
00:41901ee9
R1:8d36a012 R2:bc320a23

Internal state at time -15
15:5b6acbf6 14:10da5941 13:789c639c 12:3448fcc0 11:2af10db2 10:2567b94a 09:1bdc9fab
08:10e10abb
07:625ac5d7 06:643ae5a6 05:0e633ce7 04:4b7f87ed 03:481c5b9d 02:74265785 01:24b3f49f
00:411efa99
R1:92b7231b R2:9ec667b9

Internal state at time -14
15:17060ce1 14:5b6acbf6 13:10da5941 12:789c639c 11:3448fcc0 10:2af10db2 09:2567b94a
08:1bdc9fab
07:10e10abb 06:625ac5d7 05:643ae5a6 04:0e633ce7 03:4b7f87ed 02:481c5b9d 01:74265785
00:24b3f49f
R1:538a936d R2:c036bc48

Internal state at time -13
15:35368174 14:17060ce1 13:5b6acbf6 12:10da5941 11:789c639c 10:3448fcc0 09:2af10db2
08:2567b94a
07:1bdc9fab 06:10e10abb 05:625ac5d7 04:643ae5a6 03:0e633ce7 02:4b7f87ed 01:481c5b9d
00:74265785
R1:1a29f0af R2:7e65408e

Internal state at time -12
15:5cf4385a 14:35368174 13:17060ce1 12:5b6acbf6 11:10da5941 10:789c639c 09:3448fcc0
08:2af10db2
07:2567b94a 06:1bdc9fab 05:10e10abb 04:625ac5d7 03:643ae5a6 02:0e633ce7 01:4b7f87ed
00:481c5b9d
R1:5e4350bb R2:d0826c98

Internal state at time -11
15:479943df 14:5cf4385a 13:35368174 12:17060ce1 11:5b6acbf6 10:10da5941 09:789c639c
08:3448fcc0
07:2af10db2 06:2567b94a 05:1bdc9fab 04:10e10abb 03:625ac5d7 02:643ae5a6 01:0e633ce7
00:4b7f87ed
R1:5b20edbc R2:f327d61e

Internal state at time -10
15:2753bab2 14:479943df 13:5cf4385a 12:35368174 11:17060ce1 10:5b6acbf6 09:10da5941
08:789c639c
07:3448fcc0 06:2af10db2 05:2567b94a 04:1bdc9fab 03:10e10abb 02:625ac5d7 01:643ae5a6
00:0e633ce7
R1:2d9c405e R2:11418c75

Internal state at time -9
15:73775d6a 14:2753bab2 13:479943df 12:5cf4385a 11:35368174 10:17060ce1 09:5b6acbf6
08:10da5941
07:789c639c 06:3448fcc0 05:2af10db2 04:2567b94a 03:1bdc9fab 02:10e10abb 01:625ac5d7
00:643ae5a6
R1:972d2a15 R2:08a36a3b

Internal state at time -8
15:43930a37 14:73775d6a 13:2753bab2 12:479943df 11:5cf4385a 10:35368174 09:17060ce1
```

```
08:5b6acbf6
07:10da5941 06:789c639c 05:3448fcc0 04:2af10db2 03:2567b94a 02:1bdc9fab 01:10e10abb
00:625ac5d7
R1:8264eff1 R2:677f5747

Internal state at time -7
15:77b4af31 14:43930a37 13:73775d6a 12:2753bab2 11:479943df 10:5cf4385a 09:35368174
08:17060ce1
07:5b6acbf6 06:10da5941 05:789c639c 04:3448fcc0 03:2af10db2 02:2567b94a 01:1bdc9fab
00:10e10abb
R1:0da66493 R2:9125bf61

Internal state at time -6
15:15b2e89f 14:77b4af31 13:43930a37 12:73775d6a 11:2753bab2 10:479943df 09:5cf4385a
08:35368174
07:17060ce1 06:5b6acbf6 05:10da5941 04:789c639c 03:3448fcc0 02:2af10db2 01:2567b94a
00:1bdc9fab
R1:bbd3b2af R2:4b50ed23

Internal state at time -5
15:24ff6e20 14:15b2e89f 13:77b4af31 12:43930a37 11:73775d6a 10:2753bab2 09:479943df
08:5cf4385a
07:35368174 06:17060ce1 05:5b6acbf6 04:10da5941 03:789c639c 02:3448fcc0 01:2af10db2
00:2567b94a
R1:8b5d75ba R2:0b92f50c

Internal state at time -4
15:740c40b9 14:24ff6e20 13:15b2e89f 12:77b4af31 11:43930a37 10:73775d6a 09:2753bab2
08:479943df
07:5cf4385a 06:35368174 05:17060ce1 04:5b6acbf6 03:10da5941 02:789c639c 01:3448fcc0
00:2af10db2
R1:8ccae757 R2:ab3d746d

Internal state at time -3
15:026a5503 14:740c40b9 13:24ff6e20 12:15b2e89f 11:77b4af31 10:43930a37 09:73775d6a
08:2753bab2
07:479943df 06:5cf4385a 05:35368174 04:17060ce1 03:5b6acbf6 02:10da5941 01:789c639c
00:3448fcc0
R1:f888aec9 R2:04223414

Internal state at time -2
15:194b2a57 14:026a5503 13:740c40b9 12:24ff6e20 11:15b2e89f 10:77b4af31 09:43930a37
08:73775d6a
07:2753bab2 06:479943df 05:5cf4385a 04:35368174 03:17060ce1 02:5b6acbf6 01:10da5941
00:789c639c
R1:ee139bec R2:c6666f03

Internal state at time -1
15:7a9a1cff 14:194b2a57 13:026a5503 12:740c40b9 11:24ff6e20 10:15b2e89f 09:77b4af31
08:43930a37
07:73775d6a 06:2753bab2 05:479943df 04:5cf4385a 03:35368174 02:17060ce1 01:5b6acbf6
00:10da5941
R1:860a7dfa R2:bf0e0ffc

Internal state at time 0
15:3d4aa9e7 14:7a9a1cff 13:194b2a57 12:026a5503 11:740c40b9 10:24ff6e20 09:15b2e89f
08:77b4af31
07:43930a37 06:73775d6a 05:2753bab2 04:479943df 03:5cf4385a 02:35368174 01:17060ce1
00:5b6acbf6
R1:129d8b39 R2:2d7cdce1

Internal state at time 1
15:71db1828 14:3d4aa9e7 13:7a9a1cff 12:194b2a57 11:026a5503 10:740c40b9 09:24ff6e20
08:15b2e89f
07:77b4af31 06:43930a37 05:73775d6a 04:2753bab2 03:479943df 02:5cf4385a 01:35368174
00:17060ce1
R1:ab7cf688 R2:c1598aa6

Internal state at time 2
15:258937da 14:71db1828 13:3d4aa9e7 12:7a9a1cff 11:194b2a57 10:026a5503 09:740c40b9
08:24ff6e20
```