INTERNATIONAL STANDARD

ISO/IEC 18033-4

Second edition
2011-12-15

Information technology — Security techniques — Encryption algorithms —

Part 4: Stream ciphers

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

Partie 4: Chiffrements en flot

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

Reference number
ISO/IEC 18033-4:2011(E)

© ISO/IEC 2011

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

 © ISO/IEC 2011 – All rights reserved

Contents

Page

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

© ISO/IEC 2011 – All rights reserved

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 18033-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

This second edition cancels and replaces the first edition (ISO/IEC 18033-4:2005), which has been technically revised. It also incorporates the Amendment ISO/IEC 18033-4:2005/Amd.1:2009.

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

ISO/IEC 18033 consists of the following parts, under the general title *Information technology — Security techniques — Encryption algorithms*:

— *Part 1: General*

— *Part 2: Asymmetric ciphers*

— *Part 3: Block ciphers*

— *Part 4: Stream ciphers*

© ISO/IEC 2011 – All rights reserved

Introduction

This part of ISO/IEC 18033 includes stream cipher algorithms. A stream cipher is an encryption mechanism that uses a keystream to encrypt a plaintext in a bitwise or a block-wise manner. There are two types of stream ciphers: a synchronous stream cipher, in which the keystream is generated from only the secret key (and an initialization vector) and a self-synchronizing stream cipher, in which the keystream is generated from the secret key and some past ciphertexts (and an initialization vector). This part of ISO/IEC 18033 describes both pseudorandom number generators for producing keystream and output functions to combine a keystream with plaintext.

This part of ISO/IEC 18033 includes two output functions:

— Binary-additive output function; and

— MULTI-S01 output function.

This part of ISO/IEC 18033 includes five dedicated keystream generators:

— MUGI keystream generator;

— SNOW 2.0 keystream generator;

— Rabbit keystream generator;

— Decim^{v2} keystream generator; and

— KCipher-2 (K2) keystream generator.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-7a8378ae145f/iso-iec-18033-4-2011

© ISO/IEC 2011 – All rights reserved

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

Information technology — Security techniques — Encryption algorithms —

Part 4: Stream ciphers

1 Scope

This part of ISO/IEC 18033 specifies

a) output functions to combine a keystream with plaintext,

b) keystream generators for producing keystream, and

c) object identifiers assigned to dedicated keystream generators in accordance with ISO/IEC 9834.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

NOTE 1 The list of assigned object identifiers is given in Annex A.

NOTE 2 Any change to the specification of these algorithms resulting in a change of functional behaviour will result in a change of the object identifier assigned to the algorithms concerned.

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 18033-1, *Information technology — Security techniques — Encryption algorithms — Part 1: General*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 18033-1 and the following apply.

3.1
big-endian
method of storage of multi-byte numbers with the most significant bytes at the lowest memory addresses

[ISO/IEC 10118-1:2000]

3.2
ciphertext
data which has been transformed to hide its information content

[ISO/IEC 10116:2006]

© ISO/IEC 2011 – All rights reserved

3.3
confidentiality
property that information is not made available or disclosed to unauthorized individuals, entities, or processes

3.4
data integrity
property that data has not been altered or destroyed in an unauthorized manner

[ISO/IEC 9797-1:2011]

3.5
decryption
reversal of a corresponding encryption

[ISO/IEC 10116:2006]

3.6
encryption
reversible transformation of data by a cryptographic algorithm to produce ciphertext, i.e., to hide the information content of the data

[ISO/IEC 9797-1:2011]

3.7
initialization value
value used in defining the starting point of an encryption process

iTeh STANDARD PREVIEW
(standards.iteh.ai)

3.8
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g., encryption, decryption, cryptographic check function computation, signature generation, or signature verification)

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

[ISO/IEC 11770-1:2010]

3.9
keystream function
function that takes as input, the current state of the keystream generator and (optionally) part of the previously generated ciphertext, and gives as output the next part of the keystream

3.10
keystream generator
state-based process (i.e., a finite state machine) that takes as input, a key, an initialization vector, and if necessary the ciphertext, and gives as output a keystream (i.e., a sequence of bits or blocks of bits) of arbitrary length

3.11
***n*-bit block cipher**
block cipher with the property that plaintext blocks and ciphertext blocks are n bits in length

[ISO/IEC 10116:2006]

3.12
next-state function
function that takes as input, the current state of the keystream generator and (optionally) part of the previously generated ciphertext, and gives as output a new state for the keystream generator

3.13
output function
function that combines the keystream and the plaintext to produce the ciphertext

NOTE This function is often bitwise XOR.

© ISO/IEC 2011 – All rights reserved

3.14
padding
appending extra bits to a data string

[ISO/IEC 10118-1:2000]

3.15
plaintext
unencrypted information

[ISO/IEC 9797-1:2011]

3.16
secret key
key used with symmetric cryptographic techniques by a specified set of entities

[ISO/IEC 11770-3:2008]

3.17
state
current internal state of a keystream generator

4 Symbols and abbreviated terms

4.1 Symbols

iTeh STANDARD PREVIEW
(standards.iteh.ai)

$0x$	Prefix for hexadecimal values.
$0^{(n)}$	n-bit variable where 0 is assigned to every bit.
AND	Bitwise logical AND operation.
$Am^{(i)}[Y]$	The Y-th bit of the register $Am^{(i)}$ in KCipher-2 (K2).
a_i	Variables in an internal state of a keystream generator.
b_i	Variables in an internal state of a keystream generator.
CFB	Cipher FeedBack mode of a block cipher.
CTR	Counter mode of a block cipher.
C_i	Ciphertext block.
D_i	64-bit constants used for MUGI.
e_K	Symmetric block cipher encryption function using secret key K.
F	Subfunction used for MUGI.
FSM	Subfunction used for SNOW 2.0.
$GF(2^n)$	Finite field of exactly 2^n elements.
$GF(2^n)[x]$	The polynomial ring over the finite field $GF(2^n)$.

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

© ISO/IEC 2011 – All rights reserved

Init	Function which generates the initial internal state of a keystream generator.
IV	Initialization vector.
IK	Internal key used for KCipher-2 (K2).
K	Key.
M	Subfunction used for MUGI.
Next	Next-state function of a keystream generator.
NLF	Nonlinear function used for KCipher-2 (K2).
n	Block length.
OFB	Output FeedBack mode of a block cipher.
OR	Bitwise logical OR operation.
Out	Output function combining keystream and plaintext in order to generate ciphertext.
P	Plaintext.
P_i	Plaintext block.
R	Additional input to Out.
S_R	Subfunction used for MUGI.
Strm	Keystream function of a keystream generator.
SUB	Lookup table used for MUGI and SNOW 2.0.
Sub_{K2}	Subfunction used for KCipher-2 (K2).
S_i	Internal state of a keystream generator.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

NOTE During normal operation of the cipher, i will increase monotonically starting from zero. However, during initialization of the ciphers, it is convenient from a notational point of view to let i take negative values and define the starting state S_0 in terms of values of S_i for i <0.

T	Subfunction used for SNOW 2.0.
Z	Keystream.
Z_i	Keystream block.
α_{MUL}	Lookup table used for SNOW 2.0.
α_{MUL0}	Lookup table with index 0 used for KCipher-2 (K2).
α_{MUL1}	Lookup table with index 1 used for KCipher-2 (K2).
α_{MUL2}	Lookup table with index 2 used for KCipher-2 (K2).

© ISO/IEC 2011 – All rights reserved

α_{MUL3} Lookup table with index 3 used for KCipher-2 (K2).

α_{inv_MUL} Inverse lookup table used for SNOW 2.0.

ρ_1 Subfunction used for MUGI.

λ_1 Subfunction used for MUGI.

$\lceil x \rceil$ The smallest integer greater than or equal to the real number x.

$\neg x$ Bitwise complement operation.

$\bullet$ Polynomial multiplication.

|| Bit concatenation.

$+_m$ Integer addition modulo 2^m.

$\oplus$ Bitwise XOR (eXclusive OR) operation.

$\otimes$ Operation of multiplication of elements in the finite field $GF(2^n)$.

EXAMPLE $C = A \otimes B$: In this operation, the finite field is represented as a selected irreducible polynomial $F(x)$ of degree n with binary coefficients, the n-bit blocks $A = (a_0,a_1,...,a_{n-1})$ and $B = (b_0,b_1,...,b_{n-1})$ (where the a_i and b_i are bits) are represented as the polynomials, $A(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + ... + a_1$ and $B(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + ... + b_0$ respectively, then let $C(x) = A(x) \bullet B(x) \bmod F(x)$, i.e., $C(x)$ is the polynomial of degree at most $n-1$ obtained by multiplying $A(x)$ and $B(x)$, dividing the result by $F(x)$, and then taking the remainder. If $C(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + ... + c_0$ (where the c_i are bits) then let C be the n-bit block $(c_0,c_1,...,c_{n-1})$.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-70ac98ae145f/iso-iec-18033-4-2011

$\boxplus$ Modular addition operation

$<<_n t$ t-bit left shift in an n-bit register.

$>>_n t$ t-bit right shift in an n-bit register.

$<<<_n t$ t-bit left circular rotation in an n-bit register.

$>>>_n t$ t-bit right circular rotation in an n-bit register.

4.2 Functions

4.2.1 Left-truncation of bits

The operation of selecting the j leftmost bits of an array A=($a_0,a_1,...,a_{m-1}$) to generate a j-bit array is written

$(j \sim A)$=($a_0,a_1,...,a_{j-1}$)

This operation is defined only when $1 \le j \le m$.

See ISO/IEC 10116:2006.

© ISO/IEC 2011 – All rights reserved

4.2.2 Shift operation

The operation *Shift* is defined as follows: Given an n-bit variable X and a k-bit variable V where $1 \le k \le n$, the effect of the shift function Shift is to produce the n-bit variable

$$Shift_k(X \mid V) = (x_k, x_{k+1}, ..., x_{n-1}, v_0, v_1, ..., v_{k-1}) \qquad (k < n)$$

$$Shift_k(X \mid V) = (v_0, v_1, ..., v_{k-1}) \qquad (k = n)$$

The effect is to shift the bits of array X left by k places, discarding $x_0, x_1, ..., x_{k-1}$ and to place the array V in the rightmost k places of X. When $k = n$ the effect is to totally replace X by V.

See ISO/IEC 10116:2006.

4.2.3 Variable $I(k)$

The variable $I(k)$ is a k-bit variable where 1 is assigned to every bit.

5 Framework for stream ciphers

This clause contains a high-level description of a framework for the stream ciphers specified in this part of ISO/IEC 18033. A detailed description of the general model for a stream cipher is provided in Clause 6. A stream cipher specified in this part of ISO/IEC 18033 is defined by the specification of the following processes:

iTeh STANDARD PREVIEW
(standards.iteh.ai)

a) The keystream generator, which may be either

   — a Synchronous keystream generator, or

   — a Self-synchronizing keystream generator.

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

NOTE 1 Block cipher modes of operation are methods by which a block cipher can be used to construct a keystream generator. These modes are standardised in ISO/IEC 10116, and the meaning of the functions used in the specification is defined in 6.2.1 and 6.2.2.

NOTE 2 Block ciphers are defined in this part of ISO/IEC 18033.

b) The output function, which may be either

   — the Binary-additive output function, or

   — the MULTI-S01 output function.

6 General models for stream ciphers

6.1 Keystream generators

6.1.1 Synchronous keystream generators

A synchronous keystream generator is a finite-state machine. It is defined by:

a) An initialization function, *Init*, which takes as input a key K and an initialization vector IV, and outputs an initial state S_0 for the keystream generator. The initialization vector should be chosen so that no two messages are ever encrypted using the same key and the same IV.

 © ISO/IEC 2011 – All rights reserved

b) A next-state function, *Next*, which takes as input the current state of the keystream generator S_i, and outputs the next state of the keystream generator S_{i+1}.

c) A keystream function, *Strm*, which takes as input a state of the keystream generator S_i, and outputs a keystream block Z_i.

When the synchronous keystream generator is first initialized, it will enter an initial state S_0 defined by:

S_0 = *Init*(IV, K).

On demand the synchronous keystream generator will, for i=0,1,...:

a) Output a keystream block Z_i = *Strm*(S_i, K).

b) Update the state of the machine S_{i+1} = *Next*(S_i, K).

Therefore to define a synchronous keystream generator it is only necessary to specify the functions *Init*, *Next* and *Strm*, including the lengths and alphabets of the key, the initialization vector, the state, and the output block.

6.1.2 Self-synchronizing keystream generators

Generation of a keystream for a self-synchronizing stream cipher is dependent only on previous ciphertexts, the key, and the initialization vector. A general model for a keystream generator for a self-synchronizing stream cipher is now defined:

iTeh STANDARD PREVIEW
(standards.iteh.ai)

a) An initialization function, *Init*, which takes as input a key K and an initialization vector IV and outputs an internal input for the keystream generator S and r dummy ciphertext blocks C_{-1}, C_{-2}, …, C_{-r}.

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-

b) A keystream function, *Strm*, that takes as input S and r ciphertext blocks C_{i-1}, C_{i-2}, …, C_{i-r}, and outputs a keystream block Z_i.

To define a self-synchronizing keystream generator it is only necessary to specify the number of feedback blocks r and the functions *Init* and *Strm*.

NOTE A self-synchronizing stream cipher differs from a synchronous stream cipher in that the keystream depends only on previous ciphertext, the initialization vector and the key, i.e., the keystream generator operates in a stateless fashion. As a result, a decryptor for such a cipher can recover from loss of synchronization after receiving sufficient ciphertext blocks. This also means that the method of keystream generation is dependent upon the selected output function *Out*, which is typically the bitwise XOR operation.

6.2 Output functions

6.2.1 General model of output function

6.2 specifies two stream cipher output functions, i.e., techniques to be used in a stream cipher to combine a keystream with plaintext to derive ciphertext.

An output function for a synchronous or a self-synchronizing stream cipher is a function *Out* that combines a plaintext block P_i, a keystream block Z_i, and some other input R if necessary to give a ciphertext block C_i ($i \geq 0$). A general model for a stream cipher output function is now defined:

Encryption of a plaintext block P_i by a keystream block Z_i is given by:

C_i = *Out*(P_i, Z_i, R),

© ISO/IEC 2011 – All rights reserved

and decryption of a ciphertext block C_i by a keystream block Z_i is given by:

$P_i = Out^{-1}(C_i, Z_i, R)$.

The output function shall satisfy that for any keystream block Z_i, plaintext block P_i, and other input R,

$P_i = Out^{-1}(Out(P_i, Z_i, R), Z_i, R)$.

6.2.2 Binary-additive output function

A binary-additive stream cipher is a stream cipher in which the keystream, plaintext, and ciphertext blocks are strings of binary digits, and the operation to combine plaintext with keystream is bitwise XOR. The operation *Out* takes two inputs and does not use any additional information R for calculation. Let n to be the bit length of P_i. This function is specified by

$Out(P_i, Z_i, R) = P_i \oplus Z_i$.

The operation Out^{-1} is specified by

$Out^{-1}(C_i, Z_i, R) = C_i \oplus Z_i$.

NOTE The binary-additive stream cipher does not provide any integrity protection for encrypted data. If data integrity is required, either the MULTI-S01 output function or a separate integrity mechanism should be used, such as a MAC, i.e., a Message Authentication Code (such mechanisms are specified in ISO/IEC 9797).

iTeh STANDARD PREVIEW
(standards.iteh.ai)

6.2.3 MULTI-S01 output function

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bcf1-4617-b03b-79ac98ac1451/iso-iec-18033-4-2011

a) General model of MULTI-S01

MULTI-S01 is an output function for a synchronous stream cipher that supports both data confidentiality and data integrity. The MULTI-S01 encryption operation is suitable for use in an online environment. However, the decryption operation of MULTI-S01 can only be performed in an offline situation, as the integrity check is only performed after receiving all the ciphertext blocks. MULTI-S01 has a security parameter n. The computation of Out depends on the choice of a field $GF(2^n)$, i.e., on the choice of an irreducible polynomial over $GF(2)$ of degree n. The MULTI-S01 function only accepts messages whose length is a multiple of n. To encrypt messages whose length is not a multiple of n, a padding mechanism *Pad*(M) is required.

NOTE 1 The redundancy R is generated in such a way that the sender and the receiver share it. R can be a fixed public value like $0x00...0$.

b) The encryption function *Out*(P, Z, R)

Input: $n \cdot u$ -bit plaintext P, keystream $Z = (Z_0, Z_1, ...)$, where Z_i are n-bit blocks, n-bit redundancy R.

Output: Ciphertext C.

1) Let t be the lowest value of i ($i \geq 0$) such that $Z_i \neq 0^{(n)}$.

2) Let $(P_0, P_1, ..., P_{u-1}) = P$, where P_i is an n-bit block.

3) Set $P_u = Z_{t+u+3}$.

4) Set $P_{u+1} = R$.

© ISO/IEC 2011 – All rights reserved

5) For each P_i, do the following calculations (for $i = 0, 1, ..., u + 1$):

— Let $W_i = P_i \oplus Z_{t+i+1}$.

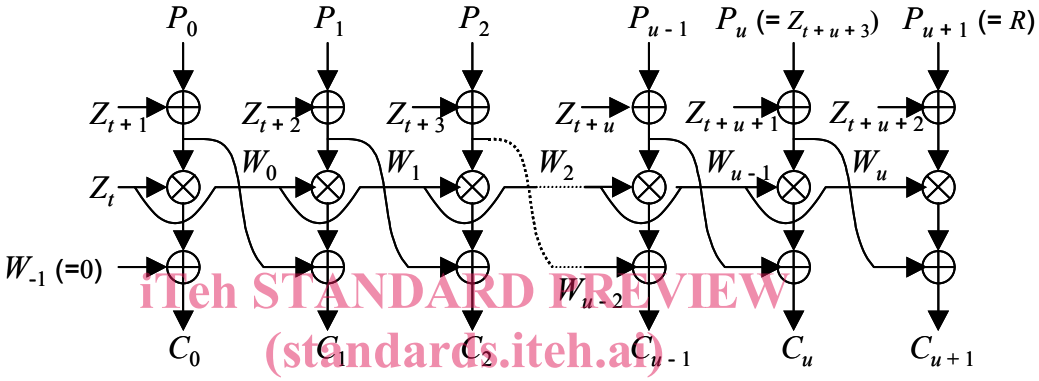
— Let $X_i = Z_t \otimes W_i$ (in $GF(2^n)$).

— Let $C_i = X_i \oplus W_{i-1}$, where W_{i-1} is the W value of the previous block $i - 1$, and $W_{-1} = 0^{(n)}$.

— Set $C = C_0 \| C_1 \| ... \| C_{u+1}$.

— Output C.

Figure 1 shows the block diagram of *Out* function.



iTeh STANDARD PREVIEW
(standards.iteh.ai)

Figure 1 — *Out* function of MULTI-S01 mode

ISO/IEC 18033-4:2011
https://standards.iteh.ai/catalog/standards/sist/5a589dc3-bc11-4617-b03b-79ac98ae145f/iso-iec-18033-4-2011

NOTE 2 The irreducible polynomial used to define multiplication in the field depends on n. For instance, in the case of $n = 64$ and 128, the irreducible polynomial $x^{64} + x^4 + x^3 + x + 1$ and $x^{128} + x^7 + x^2 + x + 1$ can be used.

c) The decryption function $Out^{-1}(P, Z, R)$

Input: $n \cdot v$-bit ciphertext C, keystream Z, n-bit redundancy R.

Output: Plaintext P or "*reject*".

1) Let t be the lowest value of i ($i \geq 0$) such that $Z_i \neq 0^{(n)}$.

2) Let $(C_0, C_1, ..., C_{v-1}) = C$, where C_i is an n-bit block.

3) For each C_i, do the following calculations (for $i = 0, 1, ..., v - 1$):

— Let $X_i = C_i \oplus W_{i-1}$, where $W_{-1} = 0^{(n)}$.

— Let $W_i = Z_t^{-1} \otimes X_i$ (in $GF(2^n)$).

— Let $P_i = W_i \oplus Z_{t+i+1}$.

4) If $P_{v-2} = Z_{t+v+1}$ and $P_{v-1} = R$, output $P = P_0 \| P_1 \| ... \| P_{v-3}$ as plaintext. Otherwise, output the special symbol meaning "*reject*" without any text.

© ISO/IEC 2011 – All rights reserved