



BSI Standards Publication

Information technology — Security techniques — Modes of operation for an n -bit block cipher

National foreword

This British Standard is the UK implementation of ISO/IEC 10116:2017+A1:2021. It supersedes [BS ISO/IEC 10116:2017](#), which is withdrawn.

The start and finish of text introduced or altered by amendment is indicated in the text by tags. Tags indicating changes to ISO/IEC text carry the number of the ISO/IEC amendment. For example, text altered by ISO/IEC amendment 1 is indicated by **A1** **A1**.

The UK participation in its preparation was entrusted to Technical Committee IST/33, Information security, cybersecurity and privacy protection.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2021
Published by BSI Standards Limited 2021

ISBN 978 0 539 05581 8

ICS 35.030

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2017.

Amendments/corrigenda issued since publication

Date	Text affected
30 June 2021	Implementation of ISO/IEC amendment 1:2021

INTERNATIONAL
STANDARD

ISO/IEC
10116

Fourth edition
2017-07-01

**Information technology — Security
techniques — Modes of operation
for an n -bit block cipher**

*Technologies de l'information — Techniques de sécurité — Modes
opératoires pour un chiffrement par blocs de n bits*



Reference number
ISO/IEC 10116:2017(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols, abbreviated terms and notation	3
4.1 Symbols and abbreviated terms	3
4.2 Notation	5
5 Requirements	5
6 Electronic Codebook (ECB) mode	6
6.1 Preliminaries	6
6.2 Encryption	6
6.3 Decryption	6
7 Cipher Block Chaining (CBC) mode	7
7.1 Preliminaries	7
7.2 Encryption	7
7.3 Decryption	7
7.4 Avoiding ciphertext expansion	8
7.4.1 General	8
7.4.2 Three ciphertext stealing variants of CBC	8
8 Cipher Feedback (CFB) mode	9
8.1 Preliminaries	9
8.2 Encryption	10
8.3 Decryption	11
8.4 Avoiding ciphertext expansion	11
9 Output Feedback (OFB) mode	12
9.1 Preliminaries	12
9.2 Encryption	12
9.3 Decryption	13
9.4 Avoiding ciphertext expansion	13
10 Counter (CTR) mode	14
10.1 Preliminaries	14
10.2 Encryption	14
10.3 Decryption	15
10.4 Avoiding ciphertext expansion	15
11 Counter advanced cryptographic prolongation of key material (CTR-ACPKM) mode	15
11.1 General	15
11.2 Preliminaries	16
11.3 Encryption	17
11.4 Decryption	18
11.5 ACPKM transformation $\langle A_1 \rangle$	18
Annex A (normative) Object identifiers	19
Annex B (informative) Properties of the modes of operation and important security guidance	21
Annex C (informative) Figures describing the modes of operation	28
Annex D (informative) Numerical examples for the modes of operation	34
Bibliography	49

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.

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 ^[A1] or www.iec.ch/members_experts/refdocs ^[A1]).

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) ^[A1] or the IEC list of patent declarations received (see patents.iec.ch) ^[A1].

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

For an explanation on 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 ^[A1] www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards ^[A1].

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, SC 27, *IT Security techniques*.

This fourth edition cancels and replaces the third edition ([ISO/IEC 10116:2006](#)) and ISO/IEC 10116:2006/Cor1:2008, which have been technically revised.

The main technical changes between the third edition and this fourth edition are as follows:

- a) the inclusion of padding within the normative scope of ISO/IEC 10116;
- b) the inclusion of methods for avoiding ciphertext expansion for CBC, CFB, OFB and CTR modes.

Introduction

This document specifies modes of operation for an n -bit block cipher. These modes provide methods for encrypting and decrypting data using a block cipher.

A1 This document specifies the following modes of operation:

- a) electronic codebook (ECB);
- b) cipher block chaining (CBC);
- c) cipher feedback (CFB);
- d) output feedback (OFB);
- e) counter (CTR);
- f) counter advanced cryptographic prolongation of key material (CTR-ACPKM). **A1**

This page deliberately left blank

Information technology — Security techniques — Modes of operation for an n -bit block cipher

1 Scope

[A1] This document establishes the modes of operation for applications of an n -bit block cipher (e.g. protection of data during transmission or in storage). **[A1]** The defined modes only provide protection of data confidentiality. Protection of data integrity is not within the scope of this document. Also, most modes do not protect the confidentiality of message length information.

NOTE 1 Methods for protecting the integrity of data using a block cipher are provided in [ISO/IEC 9797-1](#).

NOTE 2 Methods for simultaneously protecting the confidentiality and integrity of data are provided in [ISO/IEC 19772](#).

This document specifies the modes of operation and gives recommendations for choosing values of parameters (as appropriate).

[A1] Text deleted **[A1]**

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 18033-3](#), *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*.

[ISO/IEC 29192-2](#), *Information technology — Security techniques — Lightweight cryptography — Part 2: Block ciphers*.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

3.1

block cipher

symmetric encipherment system with the property that the encryption algorithm operates on a block of plaintext, i.e. a string of bits of a defined length, to yield a block of ciphertext

[SOURCE: ISO/IEC 18033-1:2015, 2.9]

3.2

ciphertext

data which has been transformed to hide its information content

[SOURCE: ISO/IEC 18033-1:2015, 2.11]

3.3 counter

$\langle A_1 \rangle$ bit array of length n bits (where n is the block size of the underlying block cipher) which is used in CTR mode and CTR-ACPKM mode $\langle A_1 \rangle$

3.4 cryptographic synchronization

co-ordination of the encryption and decryption processes

3.5 decryption

reversal of a corresponding encryption

[SOURCE: ISO/IEC 18033-1:2015, 2.16, modified]

3.6 encryption

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

[SOURCE: ISO/IEC 18033-1:2015, 2.21]

3.7 feedback buffer

FB
variable used to store input data for the encryption process

Note 1 to entry: At the starting point, *FB* has the value of *SV*.

3.8 key

sequence of symbols that controls the operation of a cryptographic transformation (e.g. encryption and decryption)

[SOURCE: ISO/IEC 18033-1:2015, 2.27, modified]

3.9 *n*-bit block cipher

block cipher with the property that plaintext blocks and ciphertext blocks are n bits in length

[SOURCE: ISO/IEC 18033-1:2015, 2.29]

3.10 padding

appending extra bits to a data string

3.11 plaintext

unencrypted information

[SOURCE: ISO/IEC 18033-1:2015, 2.30]

3.12
starting variable*SV*

variable possibly derived from some initialization value and used in defining the starting point of the modes of operation

Note 1 to entry: The "starting variable" (*SV*) used in this document is similar to (possibly identical to) the "initialization value" or "initialization vector" (*IV*) used in some other International Standards. If the starting variable referred to in this document is derived from some initialization value, then it needs to be described in any application of the modes of operation. A method for deriving a starting variable from an initializing value is not defined in this document.

A₁ 3.13
key lifetime

maximum amount of data that could be processed using this key by the particular mode of operation without loss of some proven security property

3.14
section

part of plaintext that is processed with one key before this key is transformed

3.15
section key
key used to process one section **A₁****4 Symbols, abbreviated terms and notation****4.1 Symbols and abbreviated terms**

<i>C</i>	Ciphertext (string of bits)
<i>CTR</i>	Counter value
<i>dK</i>	Decryption function of the block cipher keyed by key <i>K</i> . The decryption relation defined by the block cipher is written $P = dK(C)$ where — <i>P</i> is the plaintext block; — <i>C</i> is the ciphertext block; — <i>K</i> is the key.
<i>E</i>	Intermediate variable
<i>eK</i>	Encryption function of the block cipher keyed by key <i>K</i> . The encryption relation defined by the block cipher is written $C = eK(P)$ where — <i>P</i> is the plaintext block; — <i>C</i> is the ciphertext block; — <i>K</i> is the key.
<i>F</i>	Intermediate variable
<i>FB</i>	Feedback buffer
<i>i</i>	Iteration
<i>j</i>	Size of plaintext/ciphertext variable

K	Key
m	Number of stored ciphertext blocks
n	Plaintext/ciphertext block length for a block cipher
P	Plaintext (string of bits)
q	Number of plaintext/ciphertext variables
r	Size of feedback buffer
SV	Starting variable (see Clause 5)
X	Block cipher input block
Y	Block cipher output block
$ $	Concatenation of bit strings
$(a_1, a_2, \dots, a_m)$	A one-dimensional array of bits. For example, $A = (a_1, a_2, \dots, a_m)$ and $B = (b_1, b_2, \dots, b_m)$ are two arrays of m bits, numbered from 1 to m . All arrays of bits are written with the bit with the index 1 in the leftmost position. When interpreting a bit array as an integer, the leftmost bit shall be the most significant bit.
$\oplus$	The operation of bitwise addition, modulo 2, also known as the “exclusive or” function, is shown by the symbol $\oplus$. The operation when applied to arrays A and B of the same length is defined as $A \oplus B = (a_1 \oplus b_1, a_2 \oplus b_2, \dots, a_m \oplus b_m)$.
$\langle A_1 \rangle c$	number of bits in a counter which can be modified during incrementing in the CTR-ACP-KM mode
J	number of constants in the ACPKM transformation
$K(z)$	section key
len	length of the plaintext (in bits)
N	section size (the number of bits that are processed with one section key before this key is transformed)
s	number of sections
z	iteration for sections $\langle A_1 \rangle$

4.2 Notation

$a \bmod n$	For integers a and n , $a \bmod n$ denotes the (non-negative) remainder obtained when a is divided by n . Equivalently, if $b = a \bmod n$, then b is the unique integer satisfying: — $0 \leq b < n$, and — $(b - a)$ is an integer multiple of n .
$\sim$	The operation of selecting the j leftmost bits of an array $A = (a_1, a_2, \dots, a_m)$ to generate a j -bit array is written $(j \sim A) = (a_1, a_2, \dots, a_j)$ The operation of selecting the j rightmost bits of an array $A = (a_1, a_2, \dots, a_m)$ to generate a j -bit array is written $(A \sim j) = (a_{m-j+1}, a_{m-j+2}, \dots, a_m)$ The operation is defined only when $1 \leq j \leq m$.
$\langle A_1 \rangle a(t)$	t -bit string where the value ' a ' (0 or 1) is assigned to every bit $\langle A_1 \rangle$
S_t	Given an m -bit variable X and a t -bit variable F where $1 \leq t \leq m$, the effect of the shift function $S_t(X F)$ is to produce the m -bit variable $S_t(X F) = (x_{t+1}, x_{t+2}, \dots, x_m, f_1, f_2, \dots, f_t) \quad (t < m)$ $S_t(X F) = (f_1, f_2, \dots, f_t) \quad (t = m)$ The effect is to shift the bits of array X left by t places, discarding $x_1, x_2, \dots, x_t$, and to place the array F in the rightmost t places of X . When $t = m$ the effect is to totally replace X by F .
$\langle A_1 \rangle \lceil a \rceil$	smallest integer that is greater than or equal to $a \langle A_1 \rangle$

5 Requirements

For all of the described modes, a block cipher shall be selected from [ISO/IEC 18033-3](#) and/or [ISO/IEC 29192-2](#).

For the Electronic Codebook (ECB) mode, there are no additional parameter values that need to be selected. For the Cipher Block Chaining (CBC) mode of operation (see [Clause 7](#)), one parameter m shall be selected. For the Cipher Feedback (CFB) mode of operation (see [Clause 8](#)), three parameters r, j and k need to be selected. For the Output Feedback (OFB) mode of operation (see [Clause 9](#)) and the Counter (CTR) mode of operation (see [Clause 10](#)), one parameter j needs to be selected. $\langle A_1 \rangle$ For the counter advanced cryptographic prolongation of key material (CTR-ACPKM) mode of operation (see [Clause 11](#)), three parameters c, j and N need to be selected. $\langle A_1 \rangle$ When one of these modes of operation is used, the same parameter value(s) need to be chosen and used by all communicating parties. These parameters need not be kept secret.

All modes of operation specified in this document require the parties encrypting and decrypting a data string to share a secret key K for the block cipher in use. All modes of operation apart from the electronic Codebook (ECB) mode also require the parties to share a starting variable SV , where the length of SV will depend on the mode in use. The value of the starting variable should normally be different for every data string encrypted using a particular key. How keys and starting variables are managed and distributed is outside the scope of this document; however, important security guidance related to starting variables is provided in [Annex B](#).

$\langle A_1 \rangle$ For the ECB, CBC, CFB, OFB and CTR modes of operation, the encrypter and all potential decrypters shall agree on a padding method, unless messages to be encrypted are always a multiple of m bits ($m = n$ for ECB and CBC modes, $m = j$ for CFB, OFB and CTR modes) in length or unless the mode does not require padding. $\langle A_1 \rangle$ Such a padding method shall take a (non-empty) bit string P of arbitrary length as input and give a sequence of m -bit plaintext blocks as output. Unless the length of the plaintext message is fixed or otherwise defined by the application, then the padding method shall have the property that no two distinct input bit strings can give the same sequence of m -bit blocks as output, i.e. padded data

strings can always be uniquely unpadded. ^[A1] For the CTR-ACPKM mode of operation, padding is not used by default and the bit length of the plaintext need not be a multiple of j bits. If any padding is applied by the application that invokes the encryption, then the padding method shall be known to the application that invokes the decryption. ^[A1]

This document recommends the following padding method: the (non-empty) bit string P is right-padded with a single '1' bit. The resulting string is then right-padded with as few (possibly none) '0' bits as necessary to obtain a string whose length (in bits) is a positive integer multiple of m .

NOTE This padding method is identical to padding method 2 in both [ISO/IEC 9797-1](#) and [ISO/IEC 10118-1](#) except that it only considers non-empty bit strings.

^[A1] The modes of operation specified in this document have been assigned object identifiers in accordance with ISO/IEC 9834 (all parts). [Annex A](#) lists the object identifiers which shall be used to identify the modes of operation specified in this document.

[Annex B](#) contains comments on the properties of each mode and important security guidance.

[Annex C](#) presents figures describing the modes of operation. [Annex D](#) provides numerical examples of the modes of operation. ^[A1]

6 Electronic Codebook (ECB) mode

6.1 Preliminaries

ECB mode shall be used with an agreed padding method if the bit length of the plaintext might not be a multiple of the block cipher block length n .

The variables employed by the ECB mode of encryption are

- a) the input variables
 - 1) A plaintext string of bits P .
 - 2) A key K .
- b) the output variables, i.e. a ciphertext string of bits C .

6.2 Encryption

If a padding method is to be used, then first pad the data string P to obtain a sequence of q plaintext blocks $P_1, P_2, \dots, P_q$, each of n bits. Otherwise, simply divide the data string P into a sequence of q plaintext blocks $P_1, P_2, \dots, P_q$, each of n bits.

The ECB mode of encryption continues as follows:

$$C_i = eK(P_i) \text{ for } i = 1, 2, \dots, q.$$

6.3 Decryption

The ECB mode of decryption operates as follows:

$$P_i = dK(C_i) \text{ for } i = 1, 2, \dots, q.$$

If a padding method is not in use, then P is equal to the concatenation of $P_1, P_2, \dots, P_q$.

If a padding method is in use, apply the inverse of the agreed padding procedure to the sequence of q plaintext blocks $P_1, P_2, \dots, P_q$ to obtain the decrypted plaintext string P .

NOTE Unique removal of padding is guaranteed because of the constraint on the choice of padding method given in [Clause 5](#).

7 Cipher Block Chaining (CBC) mode

7.1 Preliminaries

CBC mode encryption and decryption as defined in 7.2 and 7.3 shall use an agreed padding method if the bit length of the plaintext might not be a multiple of the block cipher block length n . If no padding method is agreed, then the plaintext shall be a whole number of complete blocks unless it is agreed to use the approach described in 7.4.

The CBC mode of operation is defined by an interleave parameter $m > 0$, the number of independent encryption processes that could operate in parallel.

NOTE 1 The value of m can be small (typically $m = 1$) and can rarely exceed 1 024.

The variables employed by the CBC mode when being used for encryption are

- a) the input variables
 - 1) A plaintext string of bits P .
 - 2) A key K .
 - 3) A sequence of m starting variables $SV_1, SV_2, \dots, SV_m$ each of n bits. Refer to Annex B for security guidance related to the value of the starting variables.

NOTE 2 If $m = 1$ and no padding is applied, then this mode is compatible with the CBC mode described in the second edition of this document.

- b) the output variables, i.e. a ciphertext string of bits C .

7.2 Encryption

If a padding method is to be used, then first pad the data string P to obtain a sequence of q plaintext blocks $P_1, P_2, \dots, P_q$, each of n bits. Otherwise, simply divide the data string P into a sequence of q plaintext blocks $P_1, P_2, \dots, P_q$, each of n bits.

The CBC mode of encryption operates as follows:

$$C_i = eK(P_i \oplus SV_i), 1 \leq i \leq \min(m, q)$$

If $q > m$, all subsequent plaintext blocks are encrypted as:

$$C_i = eK(P_i \oplus C_{i-m}), m + 1 \leq i \leq q$$

At any time during the computation, the values of the m most recent ciphertext blocks need to be stored, e.g. in a cyclically used “feedback buffer” FB (see Figure C.2).

[A1] This procedure is shown on the Figure C.1 for $m = 1$ and on the left side of Figure C.2 for $m > 1$. **[A1]**

7.3 Decryption

The CBC mode of decryption operates as follows:

$$P_i = dK(C_i) \oplus SV_i, 1 \leq i \leq \min(m, q)$$

If $q > m$, all subsequent ciphertext blocks are decrypted as:

$$P_i = dK(C_i) \oplus C_{i-m}, m + 1 \leq i \leq q$$

At any time during the computation, the values of the m most recent ciphertext blocks need to be stored, e.g. in a cyclically used “feedback buffer” FB (see Figure C.2).

[A1] This procedure is shown on the [Figure C.1](#) for $m = 1$ and on the right side of [Figure C.2](#) for $m > 1$. **[A1]**

If a padding method is not in use, then P is equal to the concatenation of $P_1, P_2, \dots, P_q$.

If a padding method is in use, apply the inverse of the agreed padding procedure to the sequence of q plaintext blocks $P_1, P_2, \dots, P_q$ to obtain the decrypted plaintext string P .

NOTE Unique removal of padding is guaranteed because of the constraint on the choice of padding method given in [Clause 5](#).

7.4 Avoiding ciphertext expansion

7.4.1 General

In order to avoid ciphertext expansion caused by padding, it is possible to implement 'ciphertext stealing'. With this method, if the final plaintext block is partial, then the fewest '0' bits are appended so as to complete it (i.e., padding method 1 in [ISO/IEC 9797-1:2011](#)) and the resulting plaintext is encrypted as above. Any expansion that would be caused by padding bits is avoided, because bits of the penultimate ciphertext block are discarded as they can be recovered from the decryption of the final ciphertext block.

Specifically, if $m=1$ (no interleaving) and the plaintext has been padded so that the rightmost p bits of the final plaintext block P_q are the '0' padding bits, then the rightmost p bits of the penultimate ciphertext block C_{q-1} are not transmitted but can be recovered as the rightmost p bits of $dK(C_q)$.

NOTE Ciphertext stealing cannot be applied if the number of bits in the plaintext is less than n .

7.4.2 Three ciphertext stealing variants of CBC

7.4.2.1 Preliminaries

This document defines three ciphertext stealing variants (CBC_CS) of CBC mode. All three are variants of the basic CBC encryption/decryption defined above using padding method 1 in [ISO/IEC 9797-1:2011](#) [1]. These CBC_CS variants differ from the basic CBC mode only in how the two ciphertext blocks C_{q-m} and C_q are processed after encryption and before decryption.

Although the CBC_CS variants can be used when the interleave parameter $m>1$, as noted in [7.1](#), typically, $m=1$ in which case the two ciphertext blocks C_{q-m} and C_q blocks are the final two ciphertext blocks C_{q-1} and C_q . The description below applies only to the case $m=1$ but can be extrapolated to the case $m>1$.

7.4.2.2 Processing after CBC encryption (ciphertext contraction)

After CBC encryption the ciphertext $C_1 | C_2 | \dots | C_{q-1} | C_q$ is contracted (reduced in length) by the number p ($0 \leq p < n$) of padding bits. If no padding occurred ($p=0$), then no contraction occurs but, as specified below, the third CBC_CS variant still modifies the ciphertext by swapping the final two ciphertext blocks C_{q-1} and C_q :

For all CBC_CS variants define, $C_{q-1}^* = (n-p) \sim C_{q-1}$. The three CBC_CS variants are defined as follows:

- For the first CBC_CS variant, the final two ciphertext blocks $C_{q-1} | C_q$ are replaced by $C_{q-1}^* | C_q$.
- For the second CBC_CS variant, the final two ciphertext blocks $C_{q-1} | C_q$ are replaced by $C_q | C_{q-1}^*$ only if the plaintext was padded (otherwise no change is made).
- For the third CBC_CS variant, the final two ciphertext blocks $C_{q-1} | C_q$ are always replaced by $C_q | C_{q-1}^*$.

7.4.2.3 CBC decryption pre-processing (ciphertext extension)

Before CBC decryption, the received contracted ciphertext is extended to a whole number of n bit blocks.

Let the number of bits in the received ciphertext be $(q-1)n+d$ where $0 \leq d < n$ and if $d>0$ then let $p=n-d$ (note in this case that $p \neq 0$).

If $q > 1$ then define $C_{q-1}^* = (n-p) \sim C_{q-1}$.

- For the first CBC_CS variant
 - if $d=0$, then the received ciphertext is a whole number of n -bit blocks and no pre-processing is required.
 - if $d>0$, then the received ciphertext is not a whole number of n -bit blocks and pre-processing is required. In this case, the rightmost $n+d$ bits of the received ciphertext are parsed into $C_{q-1}^* \mid C_q$ where C_{q-1}^* is a d -bit block and C_q is an n -bit block and the n -bit block C_{q-1} is formed as $C_{q-1}^* \mid (dK(C_q) \sim p)$.
- For the second CBC_CS variant
 - if $d=0$, then the received ciphertext is a whole number of n -bit blocks and no pre-processing is required.
 - If $d>0$, then the received ciphertext is not a whole number of n -bit blocks and pre-processing is required. In this case, the rightmost $n+d$ bits of the received ciphertext are parsed into $C_{q-1}^* \mid C_q$, where C_{q-1}^* is a d -bit block and C_q is an n -bit block and then the n -bit block C_{q-1} is formed as $C_{q-1}^* \mid (dK(C_q) \sim p)$.
- For the third CBC_CS variant, pre-processing is always required
 - if $d=0$, then the received ciphertext is a whole number of n -bit blocks the rightmost $2n$ bits of the received ciphertext are parsed as $C_q \mid C_{q-1}$. Thus, pre-processing requires that the rightmost two blocks are swapped.
 - If $d>0$, then the received ciphertext is not a whole number of n -bit blocks. In this case, the rightmost $n+d$ bits of the received ciphertext are parsed into $C_q \mid C_{q-1}^*$, where C_{q-1}^* is a d -bit block and C_q is an n -bit block, and the n -bit block C_{q-1} is formed as $C_{q-1}^* \mid (dK(C_q) \sim p)$.

CBC decryption then decrypts the ciphertext $C_1 \mid C_2 \mid \dots \mid C_{q-1} \mid C_q$ and, if $d>0$, removes the rightmost p padding bits from the resulting plaintext blocks.

NOTE NIST/SP 800-38A defines three similar variants of ciphertext stealing (the variants differ only in the ordering of the ciphertext bits).

8 Cipher Feedback (CFB) mode

8.1 Preliminaries

CFB mode encryption and decryption as defined in 8.2 and 8.3 shall use an agreed padding method if the bit length of the plaintext might not be a multiple of the parameter j . If no padding method is agreed, then the plaintext shall be a whole number of j -bit blocks unless the approach described in 8.4 is used.

Three parameters define a CFB mode of operation:

- the size of feedback buffer, r , where $n \leq r \leq 1\,024n$
- the size of feedback variable, k , where $1 \leq k \leq n$
- the size of plaintext variable, j , where $1 \leq j \leq k$

NOTE a) $r - k$ is not constrained by n in any way, i.e. $r - k$ can be less than, equal to or greater than n . Figure C.3 shows the special case where $r - k > n$.

b) If $r = n$, then this mode is compatible with the version of CFB mode described in the first edition of this document.

c) the upper bound on r , i.e. $r \leq 1\,024n$ is chosen because it provides a realistic upper bound on the number of hardware processors.

It is recommended that CFB should be used with equal values of j and k (see [B.4.1](#)).

CFB mode shall be used with an agreed padding method if the bit length of the plaintext might not be a multiple of the parameter j .

The number of j -bit plaintext blocks (after padding if a padding method is in use) is q .

The variables employed by the CFB mode of operation when being used for encryption are

- a) the input variables
 - 1) A plaintext string of bits P .
 - 2) A key K .
 - 3) A starting variable SV of r bits. Refer to [Annex B](#) for security guidance related to the value of SV .
- b) the intermediate results
 - 1) A sequence of q block cipher input blocks $X_1, X_2, \dots, X_q$, each of n bits.
 - 2) A sequence of q block cipher output blocks $Y_1, Y_2, \dots, Y_q$, each of n bits.
 - 3) A sequence of q variables $E_1, E_2, \dots, E_q$, each of j bits.
 - 4) A sequence of $q - 1$ feedback variables $F_1, F_2, \dots, F_{q-1}$, each of k bits.
 - 5) A sequence of q feedback buffer contents $FB_1, FB_2, \dots, FB_q$ each of r bits.
- c) The output variables, i.e. a ciphertext string of bits C .

8.2 Encryption

If a padding method is to be used, then first pad the data string P to obtain a sequence of q plaintext blocks $P_1, P_2, \dots, P_q$, each of j bits. Otherwise, simply divide the data string P into a sequence of q plaintext blocks $P_1, P_2, \dots, P_q$, each of j bits.

The feedback buffer, FB , is set to its starting variable

$$FB_1 = SV$$

The operation of encrypting each plaintext variable employs the following six steps.

- a) $X_i = n \sim FB_i$ (selection of leftmost n bits of FB).
- b) $Y_i = eK(X_i)$ (use of block cipher).
- c) $E_i = j \sim Y_i$ (selection of leftmost j bits of Y_i).
- d) $C_i = P_i \oplus E_i$ (generation of ciphertext variable).
- e) $F_i = I(k - j) \mid C_i$ (generation of feedback variable).
- f) $FB_{i+1} = S_k(FB_i \mid F_i)$ (shift function on FB).

These steps are repeated for $i = 1, 2, \dots, q$, ending with step (d) on the last cycle. The procedure is shown on the left side of [Figure C.3](#). The leftmost j bits of the output block Y of the block cipher are used to encrypt the j -bit plaintext variable by modulo 2 addition. The remaining bits of Y are discarded. The plaintext and ciphertext variables have bits numbered from 1 to j .