INTERNATIONAL STANDARD

ISO/IEC 10116

Third edition
2006-02-01

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 10116:2006(E)

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

Contents

Figures

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 identfying any or all such patent rights.

ISO/IEC 10116 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcomittee SC 27, *IT Security techniques*.

This third edition cancels and replaces the second edition (ISO/IEC 10116:1997) which has been revised. Implementations that comply with ISO/IEC 10116:1997 will also comply with this third edition.

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

a) CBC mode has been extended to permit interleaving; and

b) a new mode (Counter mode) has been introduced.

Introduction

ISO/IEC 10116 specifies modes of operation for an n-bit block cipher. These modes provide methods for encrypting and decrypting data where the bit length of the data may exceed the size n of the block cipher.

This third edition of ISO/IEC 10116 specifies five modes of operation:

a) Electronic Codebook (ECB);

b) Cipher Block Chaining (CBC);

c) Cipher Feedback (CFB);

d) Output Feedback (OFB); and

e) Counter (CTR).

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

1 Scope

This International Standard establishes five modes of operation for applications of an n-bit block cipher (e.g. protection of data transmission, data storage). The defined modes only provide protection of data confidentiality. Protection of data integrity and requirements for padding the data are not within the scope of this International Standard. Also most modes do not protect the confidentiality of message length information.

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

The modes of operation specified in this International Standard have been assigned object identifiers in accordance with ISO/IEC 9834. The list of assigned object identifiers is given in Annex A. In applications in which object identifiers are used, the object identifiers specified in Annex A are to be used in preference to any other object identifiers that may exist for the mode concerned.

NOTE Annex B (informative) contains comments on the properties of each mode. Block ciphers are specified in ISO/IEC 18033-3.

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 18033-3, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers.*

3 Terms and definitions

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

3.1
block chaining
encryption of information in such a way that each block of ciphertext is cryptographically dependent upon a preceding ciphertext block.

3.2
block cipher
symmetric encryption algorithm 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.
[ISO/IEC 18033-1]

3.3
ciphertext
data which has been transformed to hide its information content.

3.4
counter
bit array of length n bits (where n is the size of the underlying block cipher) which is used in the Counter mode; its value when considered as the binary representation of an integer increases by one (modulo 2^n) after each block of plaintext is processed.

3.5
cryptographic synchronization
co-ordination of the encryption and decryption processes.

3.6
decryption
reversal of a corresponding encryption.
[ISO/IEC 18033-1]

3.7
encryption
(reversible) transformation of data by a cryptographic algorithm to produce ciphertext, i.e., to hide the information content of the data.
[ISO/IEC 18033-1]

3.8
feedback buffer (*FB*)
variable used to store input data for the encryption process. At the starting point FB has the value of SV.

3.9
key
sequence of symbols that controls the operation of a cryptographic transformation (e.g. encryption, decryption).
[ISO/IEC 18033-1]

3.10
***n*-bit block cipher**
block cipher with the property that plaintext blocks and ciphertext blocks are n bits in length.

3.11
plaintext
unencrypted information.

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 The method of deriving the starting variable from the initializing value is not defined in this International Standard. It needs to be described in any application of the modes of operation.

4 Symbols (and abbreviated terms)

C	Ciphertext block.
CTR	Counter value.
d_K	Decryption function of the block cipher keyed by key K.
E	Intermediate variable.
e_K	Encryption function of the block cipher keyed by key K.
F	Intermediate variable.
FB	Feedback buffer.
i	Iteration.
j	Size of plaintext/ciphertext variable.
K	Key.
n	Plaintext/ciphertext block length for a block cipher.
m	Number of stored ciphertext blocks.
P	Plaintext block.
q	Number of plaintext/ciphertext variables.
r	Size of feedback buffer.
SV	Starting variable.
X	Block cipher input block.
Y	Block cipher output block.
$\|$	Concatenation of bit strings.

4.1 *a* mod *n*

For integers a and n, $a \mod n$ denotes the (non-negative) remainder obtained when a is divided by n. Equivalently if $b = a \mod n$, then b is the unique integer satisfying:

— $0 \leq b < n$, and

— $(b - a)$ is an integer multiple of n

4.2 array of bits

A variable denoted by a capital letter, such as P and C, represents a one-dimensional array of bits. For example,

$$A = (a_1, a_2, \ldots, a_m) \text{ and } B = (b_1, b_2, \ldots, b_m)$$

are 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.

4.3 bitwise addition modulo 2

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

$$A \oplus B = (a_1 \oplus b_1, a_2 \oplus b_2, \ldots, a_m \oplus b_m)$$

4.4 decryption

The decryption relation defined by the block cipher is written

$$P = d_K(C)$$

where

— P is the plaintext block;
— C is the ciphertext block;
— K is the key.

4.5 encryption

The encryption relation defined by the block cipher is written

$$C = e_K(P)$$

where

— P is the plaintext block;

— C is the ciphertext block;
— K is the key.

4.6 selection of bits

The operation of selecting the j leftmost bits of an array $A = (a_1, a_2, \ldots, a_m)$ to generate a j-bit array is written

$$(j \sim A) = (a_1, a_2, \ldots, a_j)$$

The operation is defined only when $1 \leq j \leq m$.

4.7 shift operation

A "shift function" S_t is defined as follows: Given an m-bit variable X and a t-bit variable F where $1 \leq t \leq m$, the effect of the shift function $S_t(X \mid F)$ is to produce the m-bit variable

$$S_t(X \mid F) = (x_{t+1}, x_{t+2}, \ldots, x_m, f_1, f_2, \ldots, f_t) \qquad (t < m)$$
$$S_t(X \mid F) = (f_1, f_2, \ldots, f_t) \qquad (t = m)$$

The effect is to shift the bits of array X left by t places, discarding $x_1, x_2, \ldots, x_t$, 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.

4.8 $I(t)$

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

5 Requirements

For some of the described modes, padding of the plaintext variables may be required. Padding techniques, although important from a security perspective, are not within the scope of this International Standard, and throughout this standard it is assumed that any padding, as necessary, has already occurred.

NOTE Advice on the selection of a padding method for use with the CBC mode of operation is provided in Annex B.2.3.

For the Cipher Block Chaining (CBC) mode of operation (see clause 7), one parameter m needs to 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. 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 International Standard 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 (see also Annex B). How keys and starting variables are managed and distributed is outside the scope of this International Standard.

6 Electronic Codebook (ECB) mode

6.1 Preliminaries

The variables employed by the ECB mode of encryption are

a) The input variables

   1) A sequence of q plaintext blocks $P_1, P_2, \ldots, P_q$, each of n bits.
   2) A key K.

b) The output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of n bits.

6.2 Encryption

The ECB mode of encryption operates as follows:

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

6.3 Decryption

The ECB mode of decryption operates as follows:

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

7 Cipher Block Chaining (CBC) mode

7.1 Preliminaries

The CBC mode of operation is defined by an interleave parameter $m > 0$, the number of ciphertext blocks that must be stored whilst processing the mode. The value of m should be small (typically $m = 1$) and at most 1024.

NOTE The choice of 1024 as the upper limit for m is somewhat arbitrary. It is intended to provide a realistic upper bound on the number of hardware processors.

The variables employed by the CBC mode are

a) The input variables

1) A sequence of q plaintext blocks $P_1, P_2, \ldots, P_q$, each of n bits.
2) A key K.
3) A sequence of m starting variables $SV_1, SV_2, \ldots, SV_m$ each of n bits.

NOTE If $m = 1$ then this mode is compatible with the CBC mode described in the second edition of this standard (ISO/IEC 10116:1997).

b) The output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of n bits.

7.2 Encryption

The CBC mode of encryption operates as follows:

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

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

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

NOTE 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).

This procedure is shown in the left side of figure C.2.

7.3 Decryption

The CBC mode of decryption operates as follows:

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

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

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

NOTE 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).

This procedure is shown in the right side of figure C.2.

8 Cipher Feedback (CFB) mode

8.1 Preliminaries

Three parameters define a CFB mode of operation:

— the size of feedback buffer, r, where $n \leq r \leq 1024n$ and $r < qn$
— 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. $r - k$ may 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 standard (ISO/IEC 10116:1991).
c) the upper bound on r, i.e. $r \leq 1024n$ 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 clause B.3.2).

The variables employed by the CFB mode of operation are

a) The input variables

1) A sequence of q plaintext variables $P_1, P_2, \ldots, P_q$, each of j bits.
2) A key K.
3) A starting variable SV of r bits.

b) The intermediate results

1) A sequence of q block cipher input blocks $X_1, X_2, \ldots, X_q$, each of n bits.
2) A sequence of q block cipher output blocks $Y_1, Y_2, \ldots, Y_q$, each of n bits.
3) A sequence of q variables $E_1, E_2, \ldots, E_q$, each of j bits.
4) A sequence of $q - 1$ feedback variables $F_1, F_2, \ldots, F_{q-1}$, each of k bits.
5) A sequence of q feedback buffer contents $FB_1, FB_2, \ldots, FB_q$ each of r bits.

c) The output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of j bits.

8.2 Encryption

The feedback buffer FB is set to its initial value

$$FB_1 = SV$$

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) $Y_i = e_K(X_i)$ (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 $i = 1, 2, ..., q$, ending with step (d) on the last cycle. The procedure is shown in 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.

The ciphertext variable is augmented by placing $k - j$ one bits in its leftmost bit positions to become the k-bit feedback variable F. Then the bits of the feedback buffer FB are shifted left by k places and F is inserted in the rightmost k places, to produce the new value of the feedback buffer FB. In this shift operation, the leftmost k bits of FB are discarded. The new n leftmost bits of FB are used as the next input X of the encryption process.

8.3 Decryption

The variables employed for decryption are the same as those employed for encryption.

The feedback buffer FB is set to its initial value

$$FB_1 = SV$$

The operation of decrypting each ciphertext variable employs the following six steps.

a) $X_i = n \sim FB_i$ (Selection of leftmost n bits of FB).
b) $Y_i = e_K(X_i)$ (Use of block cipher).
c) $E_i = j \sim Y_i$ (Selection of leftmost j bits of Y_i).
d) $P_i = C_i \oplus E_i$ (Generation of plaintext 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 $i = 1, 2, ..., q$, ending with step (d) on the last cycle. The procedure is shown in the right side of figure C.3. The leftmost j bits of the output block Y of the block cipher are used to decrypt the j-bit ciphertext variable by modulo 2 addition. The remaining bits of Y are discarded. The plaintext and ciphertext variables have bits numbered from 1 to j.

The ciphertext variable is augmented by placing $k - j$ one bits in its leftmost bit positions to become the k-bit feedback variable F. Then the bits of the feedback buffer FB are shifted left

by k places and F is inserted in the rightmost k places to produce the new value of FB. In this shift operation, the leftmost k bits of FB are discarded. The new n leftmost bits of FB are used as the next input X of the decryption process.

9 Output Feedback (OFB) mode

9.1 Preliminaries

The OFB mode of operation is defined by one parameter, i.e. the size of the plaintext variable j, where $1 \leq j \leq n$.

The variables employed by the OFB mode of operation are the

a) input variables where

   1) A sequence of q plaintext variables $P_1, P_2, \ldots, P_q$, each of j bits;
   2) A key K; and
   3) A starting variable SV of n bits;

b) intermediate results where

   1) A sequence of q block-cipher input blocks $X_1, X_2, \ldots, X_q$, each of n bits;
   2) A sequence of q block-cipher output blocks $Y_1, Y_2, \ldots, Y_q$, each of n bits; and
   3) A sequence of q variables $E_1, E_2, \ldots, E_q$, each of j bits; and

c) output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of j bits.

9.2 Encryption

The input block X is set to its initial value

$$X_1 = SV$$

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

a) $Y_i = e_K(X_i)$ (Use of block cipher).
b) $E_i = j \sim Y_i$ (Selection of leftmost j bits).
c) $C_i = P_i \oplus E_i$ (Generation of ciphertext variable).
d) $X_{i+1} = Y_i$ (Feedback operation).

These steps are repeated for $i = 1, 2, ..., q$, ending with step (c) on the last cycle. The procedure is shown on the left side of figure C.4. The plaintext and ciphertext variables have bits numbered from 1 to j.

The result of each use of the block cipher is Y_i and this is fed back to become the next value of X, namely X_{i+1}. The leftmost j bits of Y_i are used to encrypt the input variable.

9.3 Decryption

The variables employed for decryption are the same as those employed for encryption.

The input block X is set to its initial value

$$X_1 = SV$$

The operation of decrypting each ciphertext variable employs the following four steps.

a) $Y_i = e_K(X_i)$ (Use of block cipher).
b) $E_i = j \sim Y_i$ (Selection of leftmost j bits).
c) $P_i = C_i \oplus E_i$ (Generation of plaintext variable).
d) $X_{i+1} = Y_i$ (Feedback operation).

These steps are repeated for $i = 1, 2, \ldots, q$, ending with step (c) on the last cycle. The procedure is shown in the right side of figure C.4. The plaintext and ciphertext variables have bits numbered from 1 to j.

The result of each use of the block cipher is Y_i and this is fed back to become the next value of X, namely X_{i+1}. The leftmost j bits of Y_i are used to decrypt the input variable.

10 Counter (CTR) mode

10.1 Preliminaries

The Counter mode of operation is defined by one parameter, i.e. the size of plaintext variable, j, where $1 \leq j \leq n$.

The variables employed by the Counter mode of operation are the

a) input variables where

   1) A sequence of q plaintext variables $P_1, P_2, \ldots, P_q$, each of j bits;
   2) A key K; and
   3) A starting variable SV of n bits;

b) intermediate results where

   1) A sequence of q block cipher input blocks $CTR_1, CTR_2, \ldots, CTR_q$, each of n bits;

2) A sequence of q block cipher output blocks $Y_1, Y_2, \ldots, Y_q$, each of n bits; and
3) A sequence of q variables $E_1, E_2, \ldots, E_q$, each of j bits; and

c) output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of j bits.

10.2 Encryption

The counter CTR is set to its initial value

$$CTR_1 = SV$$

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

a) $Y_i = e_K(CTR_i)$ (Use of block cipher).
b) $E_i = j \sim Y_i$ (Selection of leftmost j bits of Y_i).
c) $C_i = P_i \oplus E_i$ (Generation of ciphertext variable).
d) $CTR_{i+1} = (CTR_i + 1) \mod 2^n$ (Generation of a new counter value CTR).

These steps are repeated for $i = 1, 2, ..., q$, ending with step (c) on the last cycle. The procedure is shown on the left side of figure C.5. The plaintext and ciphertext variables have bits numbered from 1 to j.

The counter value is encrypted to give an output block Y_i and the leftmost j bits of this output block Y_i are used to encrypt the input value. The counter CTR then increases by one (modulo 2^n) to produce a new counter value.

10.3 Decryption

The variables employed for decryption are the same as those employed for encryption.

The counter CTR is set to its initial value

$$CTR_1 = SV$$

The operation of decrypting each ciphertext variable employs the following four steps.

a) $Y_i = e_K(CTR_i)$ (Use of block cipher).
b) $E_i = j \sim Y_i$ (Selection of leftmost j bits of Y_i).
c) $P_i = C_i \oplus E_i$ (Generation of plaintext variable).
d) $CTR_{i+1} = (CTR_i + 1) \mod 2^n$ (Generation of a new counter value CTR).

These steps are repeated for $i = 1, 2, \ldots, q$, ending with step (c) on the last cycle. The procedure is shown in the right side of figure C.5. The plaintext and ciphertext variables have bits numbered from 1 to j.

The counter value is encrypted to give an output block Y_i and the leftmost j bits of this output block Y_i are used to encrypt the input value. The counter CTR then increases by one (modulo 2^n) to produce a new counter value.

Annex A
(normative)
Object identifiers

This annex lists the object identifiers assigned to algorithms specified in this standard.

```
ModesOfOperation {
   iso(1) standard(0) modes-of-operation(10116) single-part(0)
        asn1-module(0) algorithm-object-identifiers(0) }

DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- EXPORTS All; --

IMPORTS
        BlockAlgorithms
                FROM EncryptionAlgorithms-3 { iso(1) standard(0)
                        encryption-algorithms(18033) part(3)
                        asn1-module(0) algorithm-object-identifiers(0) };

OID ::= OBJECT IDENTIFIER -- Alias

-- Synonyms --

is10116 OID ::= { iso(1) standard(0) modes-of-operation(10116) single-part(0)}

id-mode OID ::= { is10116 mode(1) }
id-pad  OID ::= { is10116 pad(2) }

id-pad-null     RELATIVE-OID ::= { 0 } -- no padding algorithm identified
id-pad-1        RELATIVE-OID ::= { 1 } -- padding according to method specified
                                       -- in annex B, clause B.2.3

id-mode-ecb OID ::= { id-mode ecb(1) }
id-mode-cbc OID ::= { id-mode cbc(2) }
id-mode-cfb OID ::= { id-mode cfb(3) }
id-mode-ofb OID ::= { id-mode ofb(4) }
id-mode-ctr OID ::= { id-mode ctr(5) }


-- Algorithm specifications --

ModeOfOperation ::= AlgorithmIdentifier {{ ModeOfOperationAlgorithms }}

ModeOfOperationAlgorithms ALGORITHM ::= {
        { OID id-mode-ecb PARMS EcbParameters } |
        { OID id-mode-cbc PARMS CbcParameters } |
        { OID id-mode-cfb PARMS CfbParameters } |
        { OID id-mode-ofb PARMS OfbParameters } |
        { OID id-mode-ctr PARMS CtrParameters } ,
        ... -- expect additional algorithms --
}
```



```
PadAlgo ::= CHOICE {
        specifiedPadAlgo        RELATIVE-OID,
        generalPadAlgo          OID
}

EcbParameters ::= SEQUENCE {
        bcAlgo  BlockCipher     OPTIONAL,
        padAlgo PadAlgo         DEFAULT specifiedPadAlgo:id-pad-null
}

CbcParameters ::= SEQUENCE {
        m       INTEGER         DEFAULT 1,
        bcAlgo  BlockCipher     OPTIONAL,
        padAlgo PadAlgo         DEFAULT specifiedPadAlgo:id-pad-1
}

CfbParameters ::= SEQUENCE {
        r       INTEGER,        -- n<=r<=1024n where n is the cipher block length
        k       INTEGER,        -- 1<=k<=n
        j       INTEGER,        -- 1<=j<=k
        bc      BlockCipher     OPTIONAL,
        padAlgo PadAlgo         DEFAULT specifiedPadAlgo:id-pad-null
}

OfbParameters ::= SEQUENCE {
        j       INTEGER,        -- 1<=j<=n where n is the cipher block length
        bc      BlockCipher     OPTIONAL,
        padAlgo PadAlgo         DEFAULT specifiedPadAlgo:id-pad-null
}

CtrParameters ::= SEQUENCE {
        j       INTEGER,        -- 1<=j<=n where n is the cipher block length
        bc      BlockCipher     OPTIONAL,
        padAlgo PadAlgo         DEFAULT specifiedPadAlgo:id-pad-null
}


-- Auxiliary definitions --

BlockCipher ::= AlgorithmIdentifier {{ BlockAlgorithms }}

-- 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 -- ModesOfOperation --
```

Annex B
(informative)
Properties of the modes of operation

As described above, these modes provide only protection of confidentiality. Proofs of security exist for the CBC mode, the CFB mode, the OFB mode and the CTR mode. The proofs assume that a block cipher cannot be distinguished from a pseudo-random function. The probability of this assumption being invalid increases dramatically as the number of processed blocks increases to $2^{n/2}$ and beyond.

B.1 Properties of the Electronic Codebook (ECB) mode of operation

B.1.1 Environment

Binary data exchanged between computers, or people, may contain repetitions or commonly used sequences. In ECB mode, identical plaintext blocks produce (for the same key) identical ciphertext blocks.

B.1.2 Properties

Properties of the ECB mode are:

a) encryption or decryption of a block can be carried out independently of the other blocks;
b) reordering of the ciphertext blocks will result in the corresponding reordering of the plaintext blocks; and
c) the same plaintext block always produces the same ciphertext block (for the same key) making it vulnerable to a "dictionary attack", where a dictionary is built up with corresponding plaintext and ciphertext blocks.

The ECB mode is, in general, not recommended for messages longer than one block. The use of ECB may only be specified in future International Standards for those special purposes where the repetition characteristic is acceptable, blocks have to be accessed individually, or blocks have to be accessed randomly.

B.1.3 Padding requirements

Only multiples of n bits can be encrypted or decrypted. Other lengths need to be padded to an n-bit boundary.

B.1.4 Error propagation

In the ECB mode, one or more bit errors within a single ciphertext block will only affect the decryption of the block in which the error(s) occur(s). Decryption of a ciphertext block with one or more error bits will result in an expected 50% error probability for each plaintext bit in the corresponding plaintext block.

B.1.5 Synchronization

If block boundaries are lost between encryption and decryption (e.g. due to loss or insertion of a ciphertext bit), synchronization between the encryption and decryption operations will be lost until the correct block boundaries are re-established. The result of all decryption operations will be incorrect while the block boundaries are lost.

B.2 Properties of the Cipher Block Chaining (CBC) mode of operation

B.2.1 Environment

The CBC mode produces the same ciphertext whenever the same plaintext is encrypted using the same key and starting variable. Users who are concerned about this characteristic need to adopt some procedure to change the start of the plaintext, the key, or the starting variable. One possibility is to incorporate a unique identifier (e.g. an incremented counter) at the beginning of each plaintext. Another technique, which may be used when encrypting records whose size should not be increased, is to use a value for the starting variable which can be computed from the record without knowing its contents (e.g. its address in random access storage).

A randomly chosen statistically unique SV is recommended. To prevent information leakage an integrity-protected secret SV is recommended.

B.2.2 Properties

Properties of the CBC mode are:

a) the chaining operation makes the ciphertext blocks dependent on the current and the preceding plaintext blocks $P_{i-m}, P_{i-2m}, P_{i-3m}, \ldots$ and therefore rearranging ciphertext blocks does not result in a rearranging of the corresponding plaintext blocks;

b) the use of different SV values prevents the same plaintext encrypting to the same ciphertext;

c) selection of $m > 1$ enables the block cipher encryption operations to be performed in parallel. With use of parallel processing hardware, such as a pipeline-oriented circuit, it facilitates high-throughput implementations;

d) decryption of any ciphertext block is possible without decrypting any of the preceding sequence of ciphertext blocks; that is, this mode has the "random access"property for ciphertext; and

e) if the block cipher can be modelled by a pseudo-random permutation, and the SV has been randomly chosen, the CBC mode is mathematically proven to be secure in the sense that the encryption leaks no computationally non-trivial information about the plaintext, see [5].

NOTE It is a common misunderstanding that CBC mode provides data integrity — it does not.

B.2.3 Padding requirements

Only multiples of n bits can be encrypted or decrypted. Other lengths need to be padded to an n-bit boundary.

In circumstances where padding is necessary, padding method 2 of ISO/IEC 9797-1 [2], also specified as padding method 2 in ISO/IEC 10118-1 [3], is recommended for use with the CBC mode of operation. This padding method is not only simple to implement, but it also resists certain attacks on CBC mode encryption [6], [7], [8], [9]. For certain other padding methods, if an attacker is able to manipulate encrypted messages, and is also able to detect whether or not an encrypted message causes a decrypting device to fail because the decryption process reveals an incorrectly formatted padded data string, then repeated probing of this type can be used to reveal information about the plaintext.

B.2.4 Error propagation

In the CBC mode, one or more bit errors within a single ciphertext block will affect the decryption of two blocks (the block in which the error occurs and the m-th block after). An error in the i-th ciphertext block has the following effect on the resulting plaintext: the i-th plaintext block will have an expected 50% error probability for each bit. The $i + m$-th plaintext block will have an error pattern equal to that in the i-th ciphertext block.

B.2.5 Synchronization

If block boundaries are lost between encryption and decryption (e.g. due to loss or insertion of a ciphertext bit), synchronization between the encryption and decryption operations will be lost until the correct block boundaries are re-established. The result of all decryption operations will be incorrect while the block boundaries are lost.

B.3 Properties of the Cipher Feedback (CFB) mode of operation

B.3.1 Environment

The CFB mode produces the same ciphertext whenever the same plaintext is encrypted using the same key and starting variable. Users who are concerned about this characteristic need to adopt some procedure to change the start of the plaintext, the key, or the starting variable. One possibility is to incorporate a unique identifier (e.g. an incremented counter) at the beginning of each CFB message. Another technique, which may be used when encrypting records whose size should not be increased, is to use a value for the starting variable which can be computed from the record without knowing its contents (e.g. its address in random access storage).

A randomly chosen statistically unique SV is recommended.

B.3.2 Properties

Properties of the CFB mode are:

a) the chaining operation makes the ciphertext variables dependent on the current and all but a certain number of immediately preceding plaintext variables. This number depends on the selection of r, k, and j. Therefore rearranging j-bit ciphertext variables does not result in a rearranging of the corresponding j-bit plaintext variables;

b) the use of different SV values prevents the same plaintext encrypting to the same ciphertext;

c) the encryption and decryption processes in the CFB mode both use only the encryption operation of the block cipher;

d) the strength of the CFB mode depends on the size of k (maximal if $j = k = n$) and the relative sizes of j, k, n and r;

> NOTE A choice of $j < k$ will result in an increased probability of repeating occurrences of values of the input blocks. Such repeated occurrences will reveal linear relations between plaintext bits.

e) use of this mode will require approximately n/j times as many block cipher encryption operations than would ECB mode, and hence selection of a small value for j will cause greater processing overheads;

f) selection of $r \geq n+k$ enables the pipelining and the continuous operation of the block cipher (assuming the use of parallel processing hardware, such as a pipeline-oriented circuit); and

g) a security proof for CFB mode is given in [4].

B.3.3 Padding requirements

Only multiples of j bits can be encrypted or decrypted. Other lengths need to be padded to a j-bit boundary. However, usually j will be chosen equal to such a size that no padding will be required, e.g. j can be modified for the last portion of the plaintext.

B.3.4 Error propagation

In the CFB mode, errors in any j-bit unit of ciphertext will affect the decryption of succeeding ciphertext until the bits in error have been shifted out of the CFB feedback buffer. An error in the i-th ciphertext variable has the following effect on the resulting plaintext: the i-th plaintext variable will have an error pattern equal to that in the i-th ciphertext variable. The succeeding plaintext variables will have an expected 50% error probability for each bit until all incorrectly received bits have been shifted out of the feedback buffer.

B.3.5 Synchronization

If j-bit boundaries are lost between encryption and decryption (e.g. due to loss or insertion of a ciphertext bit), cryptographic synchronization will be re-established r bits after j-bit boundaries are re-established. If a multiple of j bits are lost synchronization will be re-established automat-

ically after r bits. Consequently, when $j = k = 1$, the CFB mode automatically re-establishes cryptographic synchronization after the loss or insertion of any number of ciphertext bits.

B.4 Properties of the Output Feedback (OFB) mode of operation

B.4.1 Environment

The OFB mode produces the same ciphertext whenever the same plaintext is encrypted using the same key and starting variable. Moreover, in the OFB mode the same keystream (the sequence of intermediate results E_i) is produced when the same key and SV are used. Consequently, for security reasons a specific SV should be used only once for a given key.

A randomly chosen statistically unique SV is recommended.

B.4.2 Properties

Properties of the OFB mode are:

a) the use of different SV values prevents the same plaintext encrypting to the same ciphertext, by producing different keystreams;
b) the encryption and decryption processes in the OFB mode both use only the encryption operation of the block cipher;
c) the OFB mode does not depend on the plaintext to generate the keystream used to add modulo 2 to the plaintext;
d) use of this mode will require approximately n/j times as many block cipher encryption operations than would ECB mode, and hence selection of a small value for j will cause greater processing overheads; and
e) security proofs for the OFB mode are analogous to those for the CTR mode.

B.4.3 Padding requirements

Only multiples of j bits can be encrypted or decrypted. Other lengths need to be padded to a j-bit boundary. However, usually j will be chosen equal to such a size that no padding will be required, e.g. j can be modified for the last portion of the plaintext. No padding is required when a plaintext block with z bits, $z < j$ is encrypted by bitwise addition with only the first z bits of the corresponding variable E.

B.4.4 Error propagation

The OFB mode does not extend ciphertext errors in the resultant plaintext output. Every bit in error in the ciphertext causes only one bit to be in error in the decrypted plaintext.

B.4.5 Synchronization

The OFB mode is not self-synchronizing. If the two operations of encryption and decryption get out of synchronization, the system needs to be re-initialized. Such a loss of synchronism might be caused by any number of inserted or lost ciphertext bits.

Each re-initialization should use a value of SV different from the SV values used before with the same key. The reason for this is that an identical bit stream would be produced each time for the same parameters. This would be susceptible to, for example, known plaintext and known ciphertext attacks.

Also overlapping sequences of block cipher input blocks while using the same key should be prevented as these make the OFB mode vulnerable to, for example, known plaintext and known ciphertext attacks. This can be prevented by ensuring that the number of blocks processed under any one key does not approach $2^{n/2}$.

B.5 Properties of the Counter (CTR) mode of operation

B.5.1 Environment

The Counter mode produces the same ciphertext whenever the same plaintext is encrypted using the same key and starting variable. Moreover, in the Counter mode the same keystream (the sequence of intermediate results E_i) is produced when the same key and SV are used. Consequently, for security reasons a specific SV should be used only once for a given key and the SV values should be selected so that any intermediate CTR value is not used more than once for a given key.

A randomly chosen statistically unique SV is recommended.

B.5.2 Properties

Properties of the Counter mode are:

a) the use of different SV values prevents the same plaintext encrypting to the same ciphertext, by producing different keystreams;
b) the encryption and decryption processes in the Counter mode both use only the encryption operation of the block cipher;
c) the Counter mode does not depend on the plaintext to generate the keystream used to add modulo 2 to the plaintext;
d) in the Counter Mode a ciphertext block C_i can be decrypted without decrypting the ciphertext block C_{i-1}; this is known as a random-access property;
e) selection of a small value of j will require approximately n/j times of cycles compared to ECB mode and thus cause greater processing overhead; and
f) under the same assumptions about the block cipher as in clause B.2.2 the Counter mode can also be proven to be secure [5].

B.5.3 Padding requirements

Only multiples of j bits can be encrypted or decrypted. Other lengths need to be padded to a j-bit boundary. However, usually j will be chosen equal to such a size that no padding will be required, e.g. j can be modified for the last portion of the plaintext. No padding is required when a plaintext block with z bits, $z < j$ is encrypted by bitwise addition with only the first z bits of the corresponding variable E.

B.5.4 Error propagation

The Counter mode does not extend ciphertext errors in the resultant plaintext output. Every bit in error in the ciphertext causes only one bit to be in error in the decrypted plaintext.

B.5.5 Synchronization

The Counter mode is not self-synchronizing. If the two operations of encryption and decryption get out of synchronization, the system needs to be re-initialized. Such a loss of synchronism might be caused by any number of inserted or lost ciphertext bits.

Each re-initialization should use a value of SV different from the SV values used before with the same key. The reason for this is that an identical bit stream would be produced each time for the same parameters. This would be susceptible to, for example, known plaintext and known ciphertext attacks.

Also overlapping sequences of counter values while using the same key should be prevented, as these make the Counter mode also vulnerable to, for example, known plaintext and known ciphertext attacks.

Annex C
(informative)
Figures describing the modes of operation

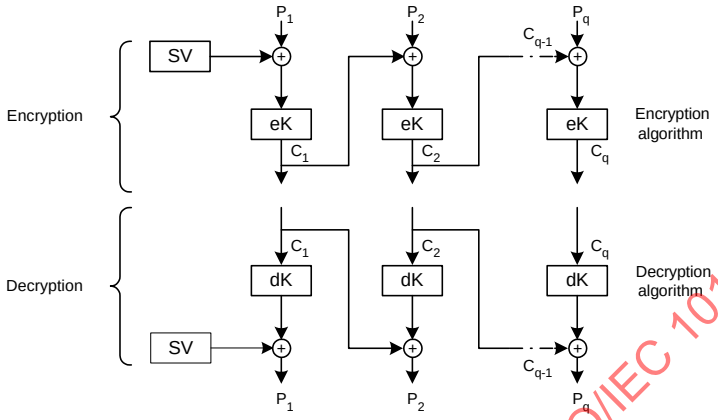

Figure C.1 – The Cipher Block Chaining (CBC) mode of operation with $m = 1$

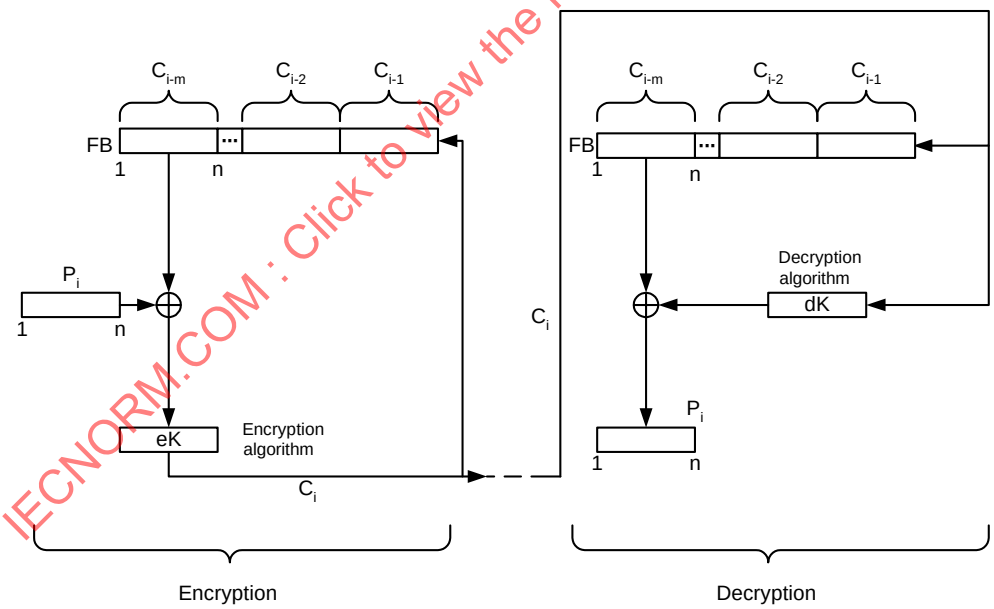

Figure C.2 – The Cipher Block Chaining (CBC) mode of operation

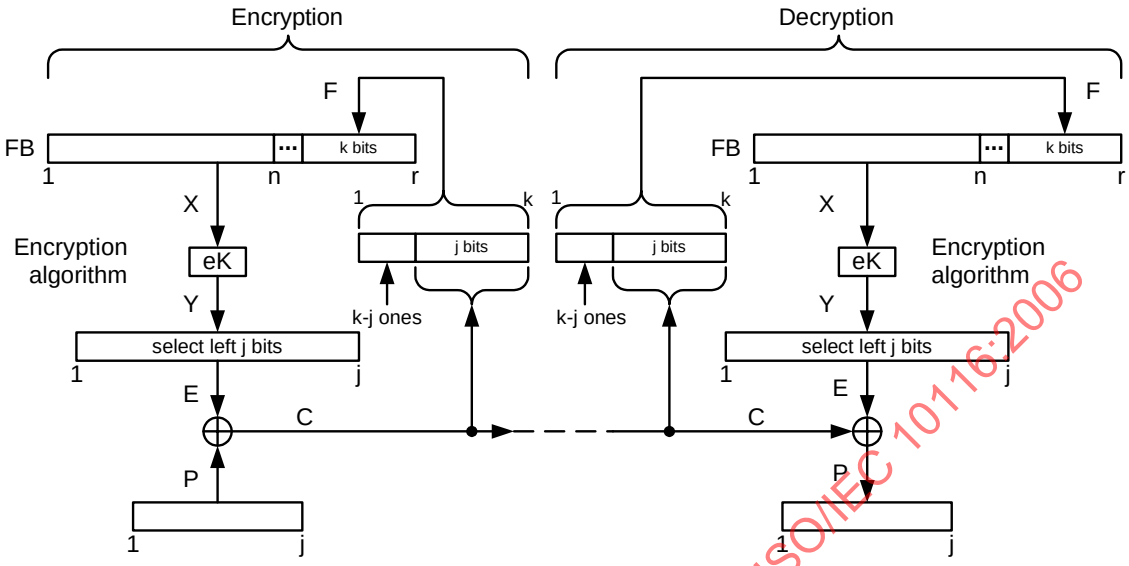

Figure C.3 – The Cipher Feedback (CFB) mode of operation

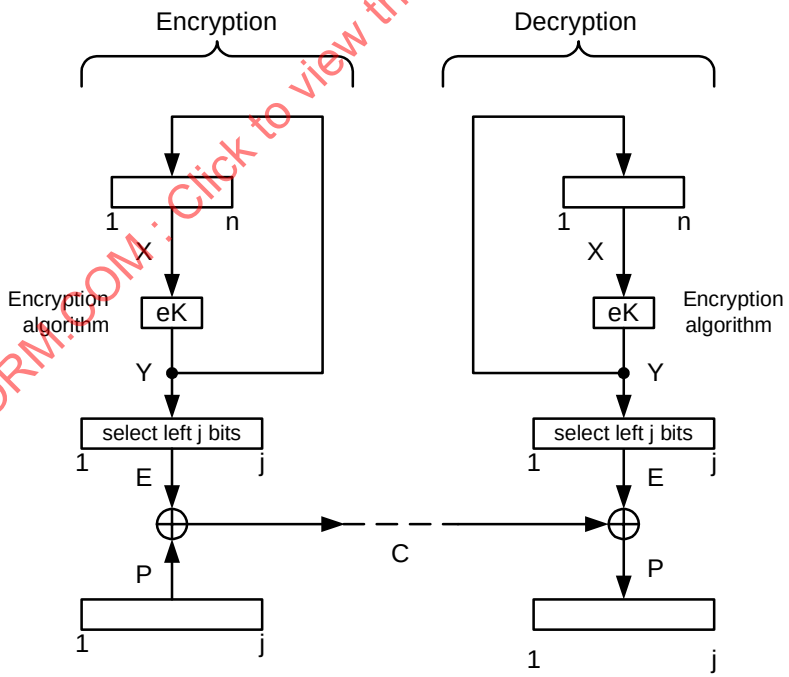

Figure C.4 – The Output Feedback (OFB) mode of operation

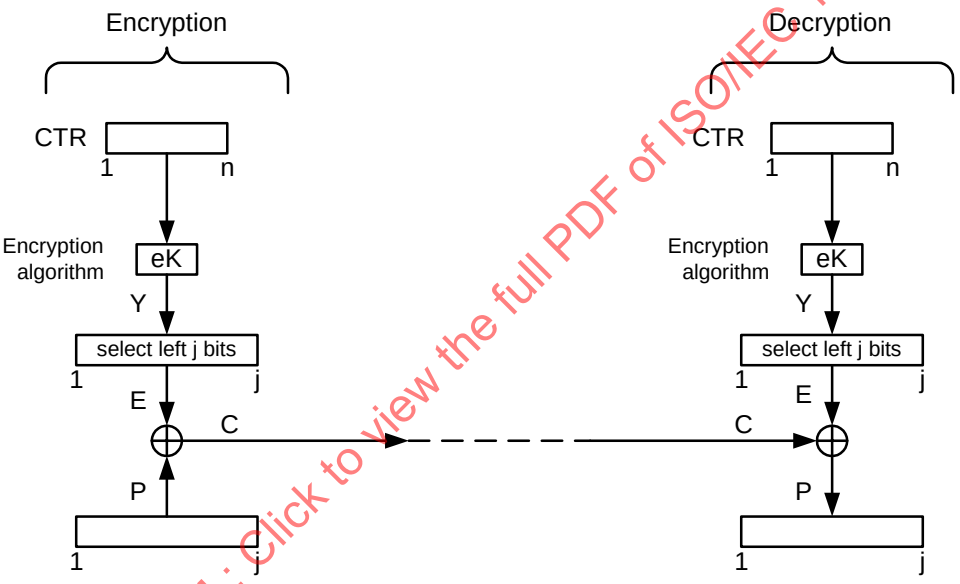

Figure C.5 – The Counter (CTR) mode of operation

Annex D
(informative)
Examples for the Modes of Operation

D.1 General

This annex gives examples for the encryption and decryption of a message using the modes of operation specified in this International Standard. The block cipher used in section D.2 is the Triple Data Encryption Algorithm (TDEA) with the value of $n = 64$. Section D.3 gives examples for the Advanced Encryption Standard (AES) with $n = 128$. The TDEA and the AES are standardized in ISO/IEC 18033-3.

D.2 Triple Data Encryption Algorithm

The examples use the following parameters:

a) The cryptographic keys are:
$K_1 =$ `0123456789ABCDEF`;
$K_2 =$ `23456789ABCDEF01`;
$K_3 =$ `456789ABCDEF0123`.

b) The starting variable is `1234567890ABCDEF`.

c) The plaintext is the 7-bit ASCII code for 'Now is the time for all ' (in hexadecimal notation `4E6F772069732074 68652074696D6520 666F7220616C6C20`). For CFB mode the plaintext is the 7-bit ASCII code for 'Now is the' (in hexadecimal notation `4E6F7720697320746865`).

The input and output of the DEA functional blocks are given sequentially. For example, in the following table, the three steps are finished sequentially in three clock cycles. If the output of DEA_1 is `3FA40E8A984D4815`, then it is the input of DEA_2, and the output of DEA_2 is `0663D1B37C48090C`, which is the input of DEA_3. The output of TDEA encryption is the output of DEA_3.

D.2.1 ECB Mode

D.2.1.1 ECB mode, encryption

Examples for the ECB mode of encryption

P_1 = 4E6F772069732074

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	4E6F772069732074	3FA40E8A984D4815
$DEA_2 - d_{K2}$	3FA40E8A984D4815	0663D1B37C48090C
$DEA_3 - e_{K3}$	0663D1B37C48090C	314F8327FA7A09A8

C_1 = 314F8327FA7A09A8

P_2 = 68652074696D6520

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	68652074696D6520	6A271787AB8883F9
$DEA_2 - d_{K2}$	6A271787AB8883F9	95CAE06A9FF4D6D2
$DEA_3 - e_{K3}$	95CAE06A9FF4D6D2	4362760CC13BA7DA

C_2 = 4362760CC13BA7DA

P_3= 666F7220616C6C20

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	666F7220616C6C20	893D51EC4B563B53
$DEA_2 - d_{K2}$	893D51EC4B563B53	5FED3EB05419F67C
$DEA_3 - e_{K3}$	5FED3EB05419F67C	FF55C5F80FAAAC45

C_3 = FF55C5F80FAAAC45

D.2.1.2 ECB mode, decryption

Examples for the ECB mode of decryption

C_1 = 314F8327FA7A09A8

	block cipher input block	block cipher output block
$DEA_1 - d_{K3}$	314F8327FA7A09A8	0663D1B37C48090C
$DEA_2 - e_{K2}$	0663D1B37C48090C	3FA40E8A984D4815
$DEA_3 - d_{K1}$	3FA40E8A984D4815	4E6F772069732074

P_1 = 4E6F772069732074

C_2 = 4362760CC13BA7DA

	block cipher input block	block cipher output block
$DEA_1 - d_{K3}$	4362760CC13BA7DA	95CAE06A9FF4D6D2
$DEA_2 - e_{K2}$	95CAE06A9FF4D6D2	6A271787AB8883F9
$DEA_3 - d_{K1}$	6A271787AB8883F9	68652074696D6520

P_2 = 68652074696D6520

C_3 = FF55C5F80FAAAC45

	block cipher input block	block cipher output block
$DEA_1 - d_{K3}$	FF55C5F80FAAAC45	5FED3EB05419F67C
$DEA_2 - e_{K2}$	5FED3EB05419F67C	893D51EC4B563B53
$DEA_3 - d_{K1}$	893D51EC4B563B53	666F7220616C6C20

P_3= 666F7220616C6C20

D.2.2 CBC Mode

D.2.2.1 CBC mode, encryption

Examples for the CBC mode of encryption with $m = 1$

$SV =$ 1234567890ABCDEF $= C_0$

$P_1 \oplus C_0 =$ 5C5B2158F9D8ED9B

	block cipher input block	block cipher output block
$\mathrm{DEA}_1 - e_{K1}$	5C5B2158F9D8ED9B	E5C7CDDE872BF27C
$\mathrm{DEA}_2 - d_{K2}$	E5C7CDDE872BF27C	EDFA50E493DBFECC
$\mathrm{DEA}_3 - e_{K3}$	EDFA50E493DBFECC	F3C0FF026C023089

$C_1 =$ F3C0FF026C023089

$P_2 \oplus C_1 =$ 9BA5DF76056F55A9

	block cipher input block	block cipher output block
$\mathrm{DEA}_1 - e_{K1}$	9BA5DF76056F55A9	5A80BE0F562EE625
$\mathrm{DEA}_2 - d_{K2}$	5A80BE0F562EE625	B92F8D3BB74CE43D
$\mathrm{DEA}_3 - e_{K3}$	B92F8D3BB74CE43D	656FBB169DEF7EDB

$C_2 =$ 656FBB169DEF7EDB

$P_3 \oplus C_2 =$ 0300C936FC8312FB

	block cipher input block	block cipher output block
$\mathrm{DEA}_1 - e_{K1}$	0300C936FC8312FB	F1F58BA6135C2B1E
$\mathrm{DEA}_2 - d_{K2}$	F1F58BA6135C2B1E	3374E4F8B29A3026
$\mathrm{DEA}_3 - e_{K3}$	3374E4F8B29A3026	30BA36075D6F0176

$C_3 =$ 656FBB169DEF7EDB

D.2.2.2 CBC mode, decryption

Examples for the CBC mode of decryption with $m = 1$

C_1 = F3C0FF026C023089

	block cipher input block	block cipher output block
$DEA_1 - d_{K1}$	F3C0FF026C023089	EDFA50E493DBFECC
$DEA_2 - e_{K2}$	EDFA50E493DBFECC	E5C7CDDE872BF27C
$DEA_3 - d_{K3}$	E5C7CDDE872BF27C	5C5B2158F9D8ED9B

$P_1 = d_{K1}(e_{K2}(d_{K3}(C_1))) \oplus C_0$ = 4E6F772069732074

C_2 = 656FBB169DEF7EDB

	block cipher input block	block cipher output block
$DEA_1 - d_{K1}$	656FBB169DEF7EDB	B92F8D3BB74CE43D
$DEA_2 - e_{K2}$	B92F8D3BB74CE43D	5A80BE0F562EE625
$DEA_3 - d_{K3}$	5A80BE0F562EE625	9BA5DF76056F55A9

$P_2 = d_{K1}(e_{K2}(d_{K3}(C_2))) \oplus C_1$ = 68652074696D6520

C_3 = 656FBB169DEF7EDB

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	30BA36075D6F0176	3374E4F8B29A3026
$DEA_2 - d_{K2}$	3374E4F8B29A3026	F1F58BA6135C2B1E
$DEA_3 - e_{K3}$	F1F58BA6135C2B1E	0300C936FC8312FB

$P_3 = d_{K1}(e_{K2}(d_{K3}(C_3))) \oplus C_2$ = 666F7220616C6C20

D.2.3 CFB Mode

D.2.3.1 CFB mode, encryption

Examples for the CFB mode of encryption. For this example the parameters $j = k = 8$ and $r = n$ have been chosen.

$SV =$ 1234567890ABCDEF

$P_1 =$ 4E

	block cipher input block	block cipher output block
$\text{DEA}_1 - e_{K1}$	1234567890ABCDEF	BD661569AE874E25
$\text{DEA}_2 - d_{K2}$	BD661569AE874E25	553A74ED589EC819
$\text{DEA}_3 - e_{K3}$	553A74ED589EC819	A011B07C73633375

$C_1 = P_1 \oplus E_1 =$ 4E $\oplus$ A0 $=$ EE

$P_2 =$ 6F

	block cipher input block	block cipher output block
$\text{DEA}_1 - e_{K1}$	34567890ABCDEFEE	A99C77AD5C012000
$\text{DEA}_2 - d_{K2}$	A99C77AD5C012000	67AC5DE460707687
$\text{DEA}_3 - e_{K3}$	67AC5DE460707687	F4FD50973F59A489

$C_2 = P_2 \oplus E_2 =$ 6F $\oplus$ F4 $=$ 9B

$P_3 =$ 77

	block cipher input block	block cipher output block
$\text{DEA}_1 - e_{K1}$	567890ABCDEFEE9B	05A76EE816060C18
$\text{DEA}_2 - d_{K2}$	05A76EE816060C18	E30ACAF870F4ED25
$\text{DEA}_3 - e_{K3}$	E30ACAF870F4ED25	73BFA8827618CC1C

$C_3 = P_3 \oplus E_3 =$ 77 $\oplus$ 73 $=$ 04

$P_4 = 20$

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	7890ABCDEFEE9B04	9CB2ADF4743DE3A6
$DEA_2 - d_{K2}$	9CB2ADF4743DE3A6	1B4FD9CE1AF12640
$DEA_3 - e_{K3}$	1B4FD9CE1AF12640	DF97078C6539370F

$C_4 = P_4 \oplus E_4 = 20 \oplus DF = FF$

$P_5 = 69$

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	90ABCDEFEE9B04FF	ED5F6A2901EA5014
$DEA_2 - d_{K2}$	ED5F6A2901EA5014	F7648978063D38CE
$DEA_3 - e_{K3}$	F7648978063D38CE	A3CC7A65EBF2124F

$C_5 = P_5 \oplus E_5 = 69 \oplus A3 = CA$

$P_6 = 73$

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	ABCDEFEE9B04FFCA	04AC08A3B0F34FFE
$DEA_2 - d_{K2}$	04AC08A3B0F34FFE	5F9FD1DDDA0AE772
$DEA_3 - e_{K3}$	5F9FD1DDDA0AE772	BD6A1C1A859057C5

$C_6 = P_6 \oplus E_6 = 73 \oplus BD = CE$

$P_7 = 20$

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	CDEFEE9B04FFCACE	73D28213BCC5750D
$DEA_2 - d_{K2}$	73D28213BCC5750D	57BB08849AAB3D72
$DEA_3 - e_{K3}$	57BB08849AAB3D72	E817358A778D6C65

$C_7 = P_7 \oplus E_7 = 20 \oplus E8 = C8$

$P_8 = 74$

	block cipher input block	block cipher output block
$DEA_1 - e_{K1}$	EFEE9B04FFCACEC8	5B88241E443B5F23
$DEA_2 - d_{K2}$	5B88241E443B5F23	E1074E8A2EFC9097
$DEA_3 - e_{K3}$	E1074E8A2EFC9097	7220B40FB6DEEC34

$C_8 = P_8 \oplus E_8 = 74 \oplus 72 = 06$

