INTERNATIONAL STANDARD

ISO/IEC 10116

Second edition
1997-04-15

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

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

Reference number
ISO/IEC 10116:1997(E)

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. 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.

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

This second edition cancels and replaces the first edition (ISO/IEC 10116:1991), which has been technically revised.

Annexes A to D of this International Standard are for information only.

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

1 Scope

This International Standard describes four modes of operation for an *n*-bit block cipher.

NOTE - Annex A (informative) contains comments on the properties of each mode.

This International Standard establishes four defined modes of operation so that in applications of an n-bit block cipher (e.g. protection of data transmission, data storage, authentication) this International Standard will provide a useful reference for, for example, the specification of the mode of operation and the values of parameters (as appropriate).

2 Definitions

For the purposes of this International Standard, the following definitions apply.

2.1 block chaining: The encipherment of information such that each block of ciphertext is cryptographically dependent upon the preceding ciphertext block.

2.2 ciphertext: Data which has been transformed to hide its information content.

2.3 cryptographic synchronization: The co-ordination of the encipherment and decipherment processes.

2.4 decipherment: The reversal of a corresponding encipherment.

2.5 encipherment: The (reversible) transformation of data by a cryptographic algorithm to produce ciphertext, i.e. to hide the data.

2.6 feedback buffer *(FB)*: Variable used to store input data for the encipherment process. At the starting point *FB* has the value of *SV*.

2.7 initializing value: Value used in defining the starting point of an encipherment process.

2.8 key: A sequence of symbols that controls the operation of a cryptographic transformation (e.g. encipherment, decipherment)

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

2.10 plaintext: Unenciphered information.

2.11 starting variable *(SV)*: Variable defining the starting point of the mode 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.

3 Notation

3.1 encipherment: For the purposes of this International Standard the functional relation defined by the block cipher is written

$$C = eK(P)$$

where

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

The expression eK is the operation of encipherment using the key K.

3.2 decipherment: The corresponding decipherment function is written

$$P = dK(C)$$

The expression dK is the operation of decipherment using the key K.

3.3 array of bits: A variable denoted by a capital letter, such as P and C above, 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 index *1* in the leftmost position.

3.4 addition modulo 2: The operation of addition, modulo 2, also known as the "exclusive or" function, is shown by the symbol ⊕. The operation applied to arrays such as A and B is defined as

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

3.5 selection of bits: The operation of selecting the j leftmost bits of A to generate a j-bit array is written

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

This operation is defined only when $1 \leq j \leq m$ where m is the number of bits in A.

3.6 shift operation: A "shift function" S_k is defined as follows:

Given an m-bit variable X and a k-bit variable F where $1 \le k \le m$, the effect of a shift function $S_k(X|F)$ is to produce the m-bit variable

$$S_k(X|F) = (x_{k+1}, x_{k+2}, ..., x_m, f_1, f_2, ..., f_k) \quad (k < m)$$
$$S_k(X|F) = (f_1, ... f_k) \quad (k = m)$$

The effect is to shift the bits of array X left by k places, discarding $x_1 ... x_k$ and to place the array F in the rightmost k places of X. When $k = m$ the effect is to totally replace X by F.

A special case of this function begins with the m-bit variable $I(m)$ of successive "1" bits and shifts the variable F of k bits into it.

The result is

$$S_k(I(m)|F) = (1, 1, ..., 1, f_1, f_2, ..., f_k) \quad (k < m)$$
$$S_k(I(m)|F) = (f_1, f_2, ..., f_k) \quad (k = m)$$

where the $m - k$ leftmost bits are "1".

4 Requirements

For some of the described modes padding of the plaintext variables may be required. Padding techniques are not within the scope of this International Standard.

For the Cipher Feedback (CFB) Mode of operation (see clause 7), three parameters r, j and k are defined. For the Output Feedback (OFB) Mode of operation (see clause 8), one parameter j is defined. When one of these modes of operation is used the same parameter value(s) need(s) to be chosen and used by all communicating parties.

5 Electronic Codebook (ECB) Mode

5.1 The variables employed for the ECB mode of encipherment are

a) A sequence of q plaintext blocks $P_1, P_2, ..., P_q$, each of n bits.
b) A key K.
c) The resultant sequence of q ciphertext blocks $C_1, C_2, ..., C_q$, each of n bits.

5.2 The ECB mode of encipherment is described as follows:

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

5.3 The ECB mode of decipherment is described as follows:

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

6 Cipher Block Chaining (CBC) Mode

6.1 The variables employed for the CBC mode of encipherment are

a) A sequence of q plaintext blocks $P_1, P_2, ..., P_q$, each of n bits.
b) A key K.
c) A starting variable SV of n bits.
d) A sequence of q ciphertext blocks $C_1, C_2, ..., C_q$, each of n bits.

6.2 The CBC mode of encipherment is described as follows:

Encipherment of the first plaintext block,

$$C_1 = eK(P_1 \oplus SV) \quad (3)$$

subsequently,

$$C_i = eK(P_i \oplus C_{i-1}) \text{ for } i = 2, 3, ..., q \quad (4)$$

This procedure is shown in the upper part of figure 1. The starting variable SV is used in the generation of the first ciphertext output. Subsequently the ciphertext is added, modulo 2, to the next plaintext before encipherment.

6.3 The CBC mode of decipherment is described as follows:

Decipherment of the first ciphertext block,

$$P_1 = dK(C_1) \oplus SV \quad (5)$$

subsequently,

$$P_i = dK(C_i) \oplus C_{i-1} \text{ for } i = 2, 3, ..., q \quad (6)$$

This procedure is shown in the lower part of figure 1.

7 Cipher Feedback (CFB) Mode

7.1 Three parameters define a CFB mode of operation:
- the size of feedback buffer, r, where $n \le r \le 2n$
- the size of feedback variable, k, where $1 \le k \le n$
- the size of plaintext variable, j, where $1 \le j \le k$

NOTES
1 $r - k$ may be smaller than n. Figure 2 shows the special case where $r - k > n$.
2 If $r = n$ then this mode is compatible with the CFB Mode described in the previous edition of this International Standard.

The variables employed for the CFB mode of operation are

a) The input variables

1) A sequence of q plaintext variables $P_1, P_2, ..., P_q$, 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 $X_1, X_2, ..., X_q$, each of n bits.
2) A sequence of q block cipher output blocks $Y_1, Y_2, ..., Y_q$, each of n bits.
3) A sequence of q variables $E_1, E_2, ..., E_q$, each of j bits.
4) A sequence of $q-1$ feedback variables $F_1, F_2, ..., F_{q-1}$, each of k bits.
5) A sequence of $q - 1$ feedback buffer contents $FB_1, FB_2, ...FB_{q-1}$, each of r bits.

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

7.2 The feedback buffer FB is set to its initial value

$$FB_1 = SV \tag{7}$$

The operation of enciphering each plaintext variable employs the following six steps:

a) $X_i = FB_i \sim n$ (8)
b) Use of block cipher, $Y_i = eK(X_i)$ (9)
c) Selection of leftmost j bits, $E_i = Y_i \sim j$ (10)
d) Generation of ciphertext variable, $C_i = P_i \oplus E_i$ (11)
e) Generation of feedback variable, $F_i = S_i(I(k)|C_i)$ (12)
f) Shift function on FB, $FB_{i+1} = S_k(FB_i|F_i)$ (13)

These steps are repeated for $i = 1, 2, ..., q$, ending with equation (11) on the last cycle. The procedure is shown in the left side of figure 2. The leftmost j bits of the output block Y of the block cipher are used to encipher 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 "1" 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 encipherment process.

7.3 The variables employed for decipherment are the same as those employed for encipherment.

The feedback buffer FB is set to its initial value

$$FB_1 = SV \tag{14}$$

The operation of deciphering each ciphertext variable employs the following six steps:

a) $X_i = FB_i \sim n$ (15)
b) Use of block cipher, $Y_i = eK(X_i)$ (16)
c) Selection of leftmost j bits, $E_i = Y_i \sim j$ (17)
d) Generation of plaintext variable, $P_i = C_i \oplus E_i$ (18)
e) Generation of feedback variable, $F_i = S_i(I(k)|C_i)$ (19)
f) Shift function on FB, $FB_{i+1} = S_k(FB_i|F_i)$ (20)

These steps are repeated for $i = 1, 2, ..., q$, ending with equation (18) on the last cycle. The procedure is shown in the right side of figure 2. The leftmost j bits of the output block Y of the block cipher are used to decipher 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 "1" 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 encipherment process.

7.4 It is recommended that CFB should be used with equal values of j and k. In this recommended form ($j = k$) the equations (12) and (19) can be written

$$F_i = C_i \quad (\text{case } j = k)$$

8 Output Feedback (OFB) Mode

8.1 The OFB 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 for the OFB mode of operation are

a) The input variables

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

b) The intermediate results

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

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

8.2 The input block X is set to its initial value

$$X_1 = SV \tag{21}$$

The operation of enciphering each plaintext variable employs the following four steps:

a) Use of block cipher, $Y_i = eK(X_i)$ (22)
b) Selection of leftmost j bits, $E_i = Y_i \sim j$ (23)
c) Generation of ciphertext variable, $C_i = P_i \oplus E_i$ (24)
d) Feedback operation, $X_{i+1} = Y_i$ (25)

These steps are repeated for $i = 1, 2, ..., q$, ending with equation (24) on the last cycle. The procedure is shown on the left side of figure 3. The result of each use of the block

cipher, which is Y_i, is used to feed back and become the next value of X, namely X_{i+1} . The leftmost j bits of Y_i are used to encipher the input variable.

8.3 The variables employed for decipherment are the same as those employed for encipherment. The input block X is set to its initial value $X_1 = SV$.

The operation of deciphering each ciphertext variable employs the following four steps:

a) Use of block cipher, $Y_i = eK(X_i)$ (26)
b) Selection of leftmost j bits, $E_i = Y_i \sim j$ (27)
c) Generation of plaintext variable, $P_i = C_i \oplus E_i$ (28)
d) Feedback operation, $X_{i+1} = Y_i$ (29)

These steps are repeated for $i = 1,2, \ldots, q$, ending with equation (28) on the last cycle. The procedure is shown in the right side of figure 3. The values X_i and Y_i are the same as those used for encipherment; only equation (28) is different.

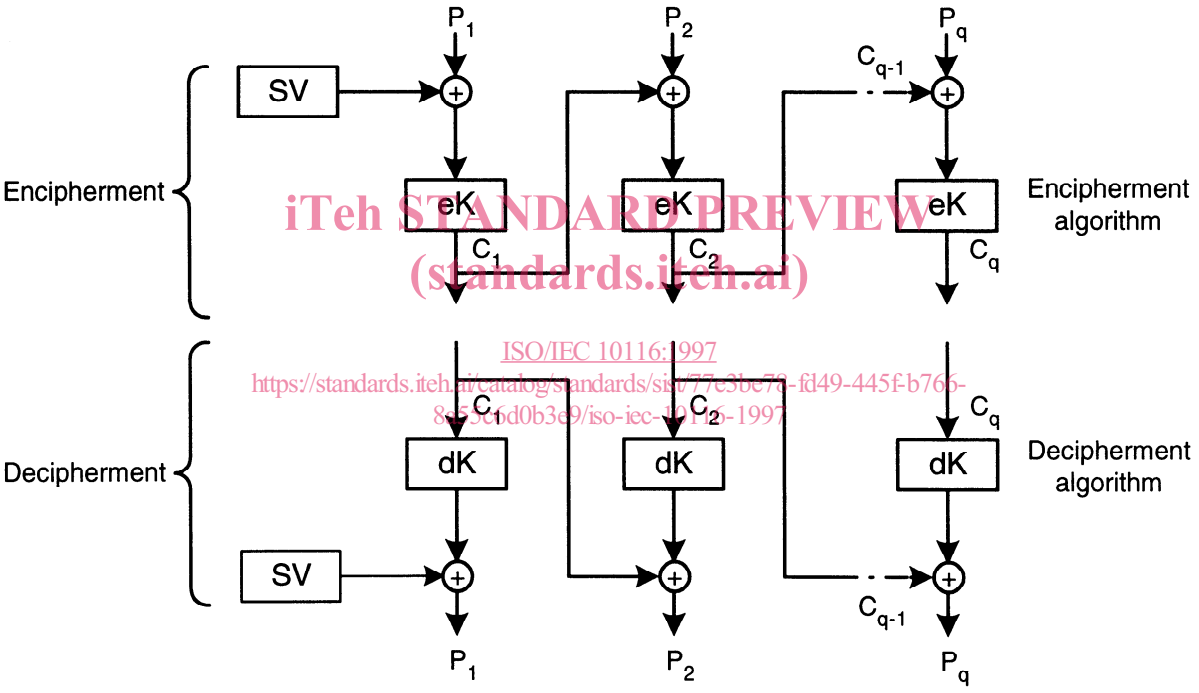

Figure 1 - The cipher block chaining (CBC) mode of operation

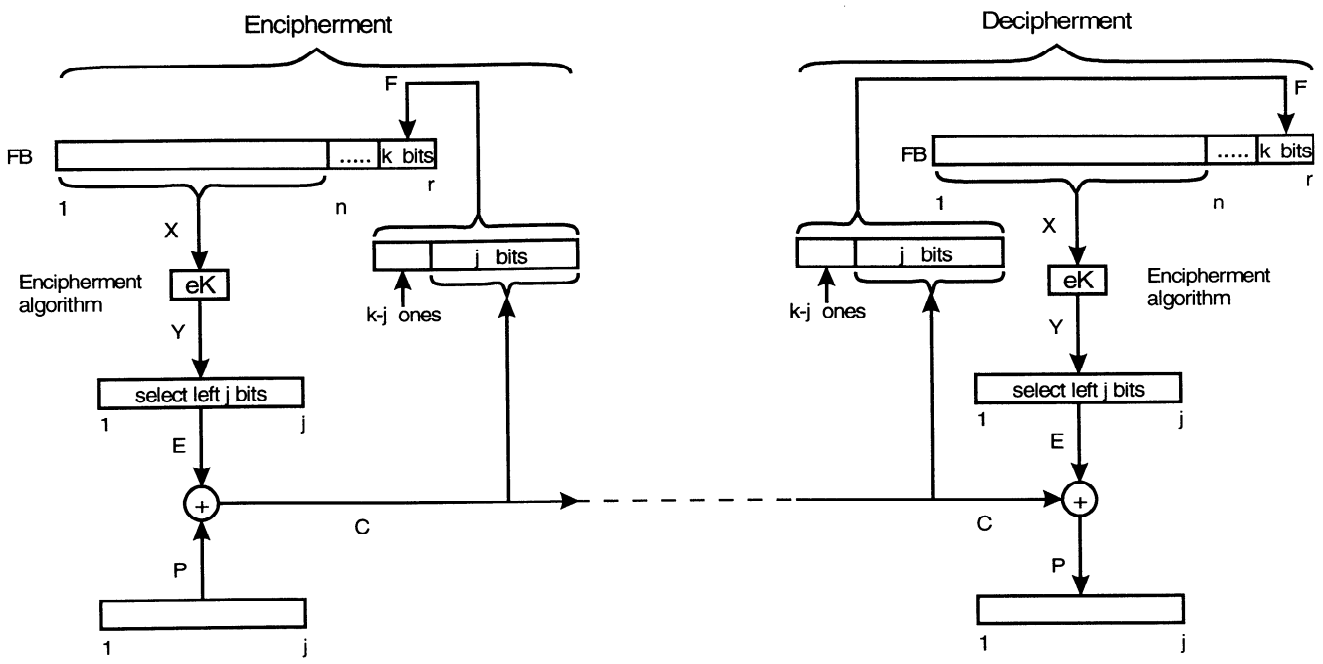

Figure 2 - The cipher feedback (CFB) mode of operation

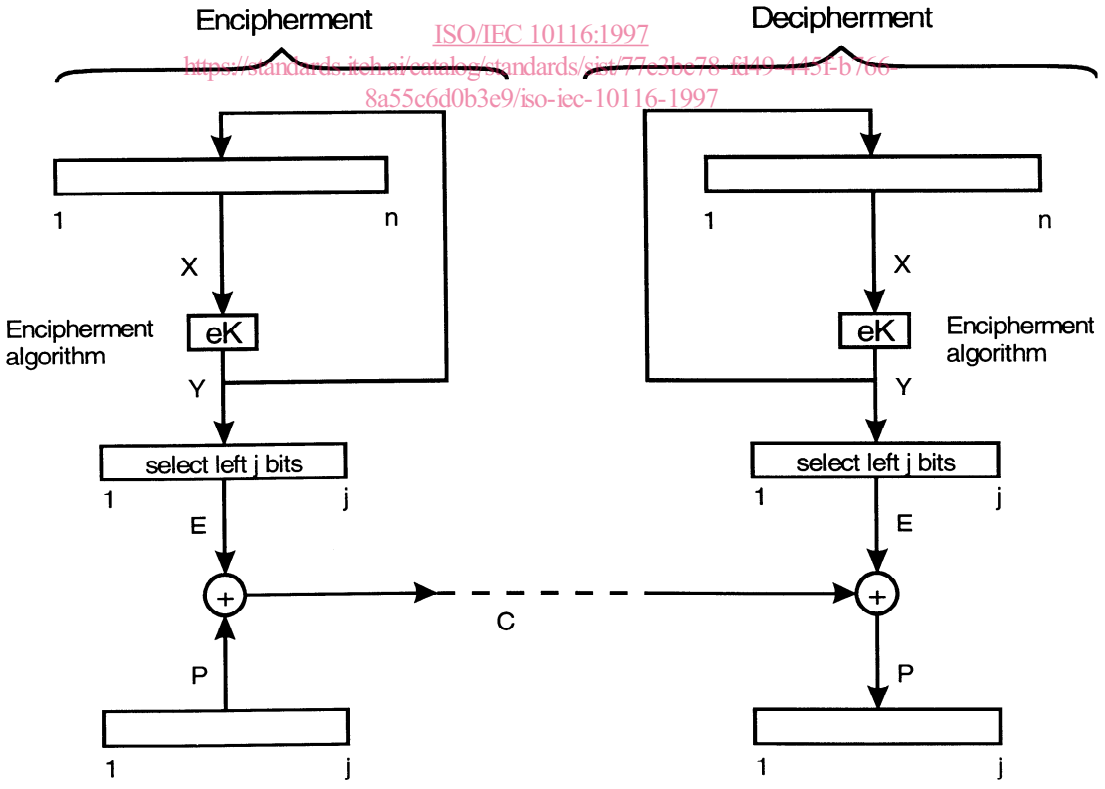

Figure 3 - The output feedback (OFB) mode of operation

Annex A
(informative)
Properties of the modes of operation

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

A.1.1 Environment

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

A.1.2 Properties

Properties of the ECB mode are

a) encipherment or decipherment 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;
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 be specified in future International Standards for those special purposes where the repetition characteristic is acceptable or blocks have to be accessed individually.

A.1.3 Padding requirements

Only multiples of n bits can be enciphered or deciphered. Other lengths need to be padded to a n-bit boundary.

A.1.4 Error propagation

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

A.1.5 Block boundaries

If block boundaries are lost between encipherment and decipherment (e.g. due to a bit slip), synchronization between the encipherment and decipherment operations will be lost until the correct block boundaries are re-established. The result of all decipherment operations will be incorrect while the block boundaries are lost.

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

A.2.1 Environment

The CBC mode produces the same ciphertext whenever the same plaintext is enciphered using the same key and starting variable. Users who are concerned about this characteristic need to adopt some ploy 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 CBC message. Another, which may be used when enciphering records whose size should not be increased, is to use some value such as the starting variable which can be computed from the record without knowing its contents (e.g. its address in random access storage).

A.2.2 Properties

Properties of the CBC mode are

a) the chaining operation makes the ciphertext blocks dependent on the current and all preceding plaintext blocks 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 enciphering to the same ciphertext.

A.2.3 Padding requirements

Only multiples of n bits can be enciphered or deciphered. Other lengths need to be padded to a n-bit boundary. If this is not acceptable, the last variable can be treated in a special way. Two examples of a special treatment are given below.

A first possibility to treat an incomplete last variable (i.e. a variable P_q of $j < n$ bits where q should be greater than 1) is to encipher it in OFB mode as described below:

a) encipherment

$$C_q = P_q \oplus (eK(C_{q-1}) \sim j) \qquad (30)$$

b) decipherment

$$P_q = C_q \oplus (eK(C_{q-1}) \sim j) \qquad (31)$$

However, this last variable is vulnerable to a "chosen plaintext attack" if the *SV* is not secret or if it is used more than once with the same key (see clause A.4).

A second possibility is known as "ciphertext-stealing". Suppose that the last two plaintext variables are P_{q-1} and P_q, where P_{q-1} is an n-bit block and P_q is a variable of $j < n$ bits and q should be greater than 1.

a) encipherment

Let C_{q-1} be the ciphertext block derived from P_{q-1} using the method described in 5.2. Then set

$$C_q = eK(S_j(C_{q-1} \mid P_q)) \quad (32)$$

The last two ciphertext variables are then $C_{q-1} \sim$ j and C_q.

b) decipherment

C_q needs to be deciphered first, resulting in the variable P_q and the right-most n-j bits of C_{q-1}

$$S_j(C_{q-1} \mid P_q) = dK(C_q) \quad (33)$$

The complete block C_{q-1} is now available and P_{q-1} can be derived using the method described in 5.3.

The two trailing ciphertext variables are deciphered in reverse order which makes this solution less suited for hardware implementations.

A.2.4 Error propagation

In the CBC mode, one or more bit errors within a single ciphertext block will affect the decipherment of two blocks (the block in which the error occurs and the succeeding block). An error in the i-th ciphertext block has the following effect on the resulting plaintext: the i-th plaintext block will have a 50 % error probability for each bit. The i+1-th plaintext block will have an error pattern equal to that in the i-th ciphertext block. If errors occur in a variable of less than n bits, error propagation depends on the chosen method of special treatment. In the first example the deciphered short block will have those bits in error that correspond directly to the ciphertext bits in error.

A.2.5 Block boundaries

If block boundaries are lost between encipherment and decipherment (e.g. due to a bit slip), synchronization between the encipherment and decipherment operations will be lost until the correct block boundaries are re-established. The result of all decipherment operations will be incorrect while the block boundaries are lost.

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

A.3.1 Environment

The CFB mode produces the same ciphertext whenever the same plaintext is enciphered using the same key and starting variable. Users who are concerned about this characteristic need to adopt some ploy 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, which may be used when enciphering records whose size should not be increased, is to use some value such as the starting variable which can be computed from the record without knowing its contents (e.g. its address in random access storage).

A.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 (see figure 2). 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 enciphering to the same ciphertext;

c) the encipherment and decipherment processes in the CFB mode both use the encipherment operation of the block cipher;

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

NOTE - $j < k$ 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) selection of a small value of j will require more cycles through the block cipher operation per unit of plaintext and thus cause greater processing overheads.

f) selection of $r \geq n + k$ enables the pipelining and the continuous operation of the block cipher.

A.3.3 Padding requirements

Only multiples of j bits can be enciphered or deciphered. Other lengths need to be padded to a j-bit boundary. However, frequently 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.

A.3.4 Error propagation

In the CFB mode, errors in any j-bit unit of ciphertext will affect the decipherment 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 a 50 % error probability for each bit until all incorrectly received bits have been shifted out of the feedback buffer.

A.3.5 Synchronization

If j-bit boundaries are lost between encipherment and decipherment (e.g. due to a bit slip), 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 automatically after r bits.