通信情報保護學會誌
第3卷 第4號, 1993. 12

ISO/IEC JTC1/SC27의 국제표준소개 (3) : ISO/IEC IS 10116

정보 기술 - n 비트 블럭 암호 알고리즘의 운영 모드

[Information technology - Modes of operation for an n-bit block cipher algorithm]

이 필 중*

이 시리즈를 시작하면서 제3권 제2호(1993.6)에 SC27 국제표준화 현황을 정리해 보고했었다. 필자와 산업연구원의 이경석 박사가 1993년 10월 파리에서 열린 SC27 국제표준 총회에 다녀와 새로 정리된 최신의 표준화현황은 11월 20일 개최되는 1993년도 한국통신정보보호학회 학술대회에서 특별보고회로 발표되며 그 내용은 논문집에 포함되어 있으니 참고 바란다.

이번 호에는 n비트의 입출력을 갖는 블럭 암호화 알고리즘의 사용방법에 관해 1991년에 국제표준이 된 문서 IS 10116를 소개한다. 이보다 앞서 1987년 IS 8372로 거의 같은 내용의 국제표준이 n=64의 경우에 한정되어 만들어져 사용되어왔다. 1992년 정보보안 국제총회에서 IS 10116이 IS 8372의 내용을 포함하며 IS 10116가 IS 8372보다 더 잘 서술되어 있으므로 IS 8372를 폐지할 것을 검토한적이 있었다. 그러나 이미 많은 제품이 IS 8372를 근거로 만들어져 있기 때문에 당분간 IS 8372를 국제표준으로 놓아두자는 결론을 냈었다.

1. 범 위 [Scope]

이 국제표준은 n비트 블럭 암호 알고리즘의 4가지 운영모드를 기술하고 있다. [This International Standard describes four modes of operation for an n-bit block cipher algorithm.]

주1 : 부록A(참고)는 각 모드의 특성에 관한 주석을 포함하고 있다. [NOTE 1 : Annex A(Informative) contains comments on the properties of each mode.]

이 국제표준은 4개의 정의된 운영모드를 규정함으로써, n비트 블럭 암호 알고리즘의 응용(즉 데이타 전송의 보호, 데이타 저장, 인증)들에 있어서, 이 국제표준은, 예를들면, 운영 모드의 규정과 파라미터 값 등에 유용한 참고자료가 될 수 있다. [This International Standard establishes four defined modes of operation so that in applications of an n-bit block cipher algorithm(e.g. protection of data

* 포항공과대학 전자전기공학과

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).]

어떤 모드에서는, 모든 평문 변수들이 필요한 길이가 되도록 하기 위하여 채워넣기가 필요하다. 채워넣기 방법은 이 국제표준의 범위에 포함되지 않는다.[For some modes, padding may be required to ensure that all plaintext variables are of the necessary length. Padding techniques are not within the scope of this International Standard.]

주 2 : 암호 피드백(CFB) 운영모드(6절 참조)에는 j와 k 두 개의 파라미터가 정의되어 있다. 출력 피드백(OFB) 운영 모드(7절 참조)에는 1개의 파라미터 j가 정의되어 있다. 이들 운영 모드 중의 하나가 사용될 때, 해당 파라미터 값(들)이 모든 통신 참여자들에 의해 선택되어지고, 사용되어질 필요가 있다.[**NOTE 2** : For the Cipher Feedback(CFB) Mode of operation(see clause 6), two parameters j and k are defined. For the Output Feedback (OFB) Mode of operation(see clause 7), one parameter j is defined. When one of these modes of operation is used the parameter value(s) need(s) to be chosen and used by all communicating parties.]

2. 정 의 [Definitions]

이 국제표준을 위해 다음과 같은 정의들이 적용된다.[For the purpose of this International Standard, the following definitions apply.]

2.1. **평문 [plaintext]** : 암호화되지 않은 정보. [Unenciphered information.]

2.2. **암호문 [ciphertext]** : 암호화된 정보. [Enciphered information.]

2.3. **n비트 블럭 암호 알고리즘 [n bit block cipher algorithm]** : 평문과 암호문이 n-비트인 블럭 암호 알고리즘. [A block cipher algorithm with the property that plaintext blocks and ciphertext blocks are n bits in length.]

2.4. **블럭 체이닝 [block chaining]** : 암호문의 각 블럭이 선행 암호문 블럭에 암호적으로 연관성이 있게 한 정보의 암호화.[The encipherment of information such that each block of ciphertext is cryptographically dependent upon the preceding ciphertext block.]

2.5. **초기값 [initializing value(IV)]** : 암호화 과정의 시작점 설정에 사용되는 값.[Value used in defining the starting point of an encipherment process.]

2.6. **시작변수 [starting variable(SV)]** : 초기값에서 유도되어 운영 모드의 시작점 설정에 사용되는 변수.[Variable derived from the initializing value and used in defining the starting point of the modes of operation.]

주 3 : 이 국제표준에서는 초기값에서 시작 변수를 유도하는 방법이 정의되어 있지 않다. 운영 모드의 어떠한 응용에서든 기술될 필요가 있다. [**NOTE 3** : 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.]

2.7. **암호동기 [cryptographic synchronization]** : 암호화와 복호화 과정의 시작점 조정.[The coordination of the encipherment and decipherment process.]

3. 표기법 [Notation]

이 국제표준에서는 블럭 암호 알고리즘에 의해 정의된 함수관계를 다음과 같이 표시한다.

$$C = eK(P)$$

여기서 P는 평문 블럭, C는 암호문 블럭, K는 열쇠이다. 이 ek라는 표현은 열쇠 K를 사용한 암호화 연산이다. [For the purpose of this International Standard the functional relation defined by the block encipherment algorithm 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.]

대응하는 복호화 함수는 다음과 같다.

$$P = dK(C)$$

대문자로 표기된 변수는, 가령 위에서의 P와 C같이, 비트들의 일차원 배열을 나타낸다. 예를 들면,

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

는 1에서 m까지 번호가 붙여진 m비트의 배열들이다. 모든 배열의 비트들은 MSB가 왼쪽에 위치해 있다. [The corresponding decipherment function is written P=dK(C). A variable denoted by a capital letter, such as P and C above, represents a one-dimensional array of bits. For example, A= $A=(a_1, a_2, \ldots, a_m)$ 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 most significant bit in the left position.]

"배타적 논리합"으로서 알려진 2진 덧셈 연산은 기호 ⊕로 표시된다. A와 B의 배열에 대한 그 연산은 다음과 같이 정의된다.

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

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

j비트 배열을 생성하기 위해 A의 왼쪽에서 j비트를 선택하는 연산은 다음과 같이 표시된다.

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

이 연산은 m이 j<m일 경우에만 정의되며 여기서 m은 A의 비트수이다. [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 j<m where m is the number of bits in A.]

"쉬프트 함수" S_k는 다음과 같이 정의된다. m비트의 변수 X와 k비트의 변수 F(단, k≤m)가 주어졌을 때, 쉬프트 함수 $S_k(X|F)$의 결과는 다음과 같은 m비트의 변수가 된다.

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

결과는 배열 X의 비트를 k만큼 좌로 쉬프트하여, $x_1 \ldots\ldots x_k$를 없애고 X의 오른쪽에서 k만큼을 배열 F와 바꾸어 놓은 것이다. 만약 k=m이면, X는 F에 의해 전체적으로 대치된다. [A "shift function" S_k is defined as follows: Given an m-bit variable X and a k-bit variable F where k≤m, the effect of a shift function $S_k(X|F)$ is to produce the m-bit variable $S_k(X|F)=(x_{k+1}, x_{k+2}, \ldots, x_m, f_1, f_2, \ldots, f_k)$ (k<m). The effect is to shift the bits of array X left by k places, discarding $x_1 \ldots\ldots x_k$ 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.]

연속하는 "1"비트들로 이루어진 m비트 변수 I(m)을 시작으로, k비트의 변수 F를 그 안으로 쉬프트한 (단, k≤m)이 함수의 특별한 경우가 사용된다. 그 결과는 다음과 같다.

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

여기서 왼쪽으로부터 m-k개의 비트들이 "1"이다.

[A special case of this function is used which begins with the m-bit variable I(m) of successive "1" bits and shifts the variable F of k bits into it, where $k \leq m$. The results is $S_k(I(m)|F)=(1, 1, \ldots, 1, f_1, f_2, \ldots, f_k)$ $(k<m)$, $S_k(I(m)|F)=(f_1, f_2, \ldots, f_k)$ $(k=m)$ where the m-k leftmost bits are "1".]

4. 전자 코드북(ECB) 모드 [Electronic Codebook (ECB) Mode]

4.1. ECB 암호화 모드에 사용되는 변수들은 아래와 같다.

a) 각기 n비트인 q개의 평문 블럭열 $P_1, P_2, \ldots, P_q$

b) 열쇠 K

c) 각기 n비트인 q개의 암호문 블럭열 $C_1, C_2, \ldots, C_q$

[The variables employed for the ECB mode of encipherment are: a) A sequence of q plaintext blocks $P_1, P_2, \ldots, P_q$, each of n bits. b) A key K. c) The resultant sequence of q ciphertext blocks $C_1, C_2, \ldots, C_q$, each of n bits.]

4.2. ECB 암호화 모드는 다음과 같이 기술된다.

$$C_i = eK(P_i) \qquad i=1, 2, \ldots, q \qquad \ldots (1)$$

[The ECB mode of encipherment is described as follows: $C_i = eK(P_i)$ for $i=1, 2, \ldots, q$]

4.3. ECB 복호화 모드는 다음과 같이 기술된다.

$$P_i = dK(C_i) \qquad i=1, 2, \ldots, q \qquad \ldots (2)$$

[The ECB mode of decipherment is described as follows: $P_i = dK(C_i)$ for $i=1, 2, \ldots, q$]

5. 암호 블럭 체이닝(CBC) 모드 [Cipher Block Chaining (CBC) Mode]

5.1. CBC 암호화 모드에 사용되는 변수들은 아래와 같다.

a) 각기 n비트인 q개의 평문 블럭열 $P_1, P_2, \ldots, P_q$

b) 열쇠 K

c) n비트의 시작 변수 SV

d) 각기 n비트인 q개의 암호문 블럭열 $C_1, C_2, \ldots, C_q$

[The variables employed for the CBC mode of encipherment are: a) A sequence of q plaintext blocks $P_1, P_2, \ldots, P_q$, each of n bits. b) A key K. c) A starting variable SV of n bits. d) A sequence of q ciphertext blocks $C_1, C_2, \ldots, C_q$, each of n bits.]

5.2. CBC 암호화 모드는 다음과 같이 기술된다.

최초의 평문 블럭의 암호화 ;

$$C_1 = eK(P_1 \oplus SV) \qquad \ldots (3)$$

계속하여 ;

$$C_i = eK(P_i \oplus C_{i-1}) \qquad i=2, 3, \ldots, q \qquad \ldots (4)$$

[The CBC mode of encipherment is described as follows: Encipherment of the first plaintext block, $C_1 = eK(P_1 \oplus SV)$ ... (3) subsequently, $C_i = eK(P_i \oplus C_{i-1})$ ... (4) for $i=2, 3, \ldots, q$]

이 절차는 그림 1의 상부에 나타나 있다. 시작변수 SV는 최초의 암호문 출력을 만들때 사용된다. 그 다음부터는 암호문을 2진 덧셈 연산으로 암호화되기 전의 다음 평문과 더해진다. [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.]

5.3. CBC 복호화 모드는 다음과 같이 기술된다.

최초의 평문 블럭의 복호화 ;

$$P_1 = dK(C_1 \oplus SV) \qquad \ldots (5)$$

계속하여 ;

$$P_i = dK(C_i \oplus C_{i-1}) \qquad i=2, 3, \ldots, q \qquad \ldots (6)$$

이 절차는 그림 1의 하부에 나타나 있다. [The CBC mode of decipherment is described as follows: Decipherment of the first ciphertext block, $P_1 =$

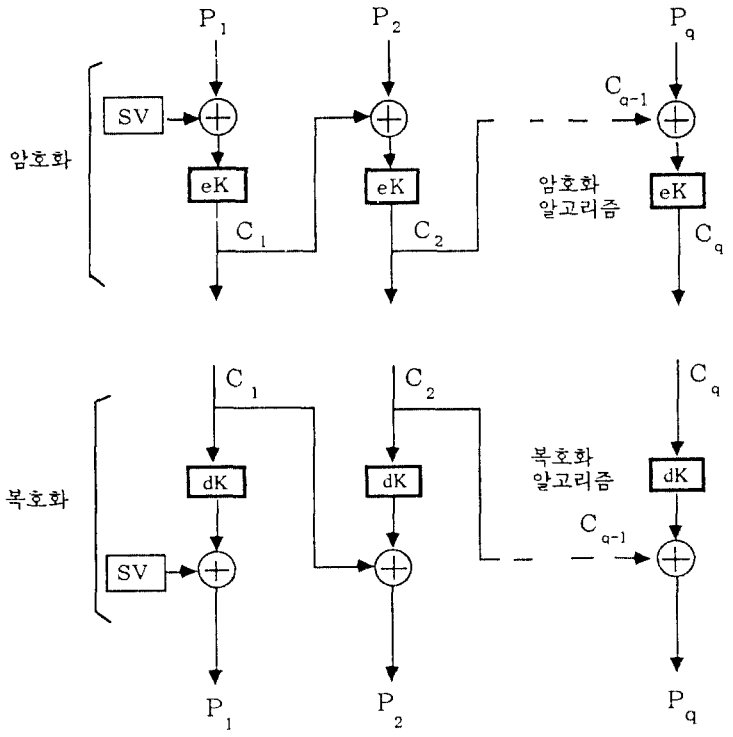


그림 1. 암호 블럭 (CBC) 체이닝 운영 모드

$dK(C_1 \oplus SV)$ ...(5) subsequently, $P_i = dK(C_i \oplus C_{i-1})$ for i=2, 3, ..., q ...(6). This procedure is shown in the lower part of figure 1.]

6. 암호 피드백(CFB) 모드 [Cipher Feedback (CFB) Mode]

6.1. 두개의 파라미터가 CFB 운영 모드를 정의한다.

- 피드백 변수의 크기, k ($1 \le k \le n$)
- 평분 변수의 크기, j ($1 \le j \le k$)

[Two parameters define a CFB mode of operation: the size of feedback variable, k, where $1 \le k \le n$, the size of plaintext variable, j, where $1 \le j \le k$.]

CFB 운영 모드에 이용되는 변수들은 아래와 같다.

a) 입력변수

① 각기 j비트인 q개의 평문 변수들 $P_1, P_2, \ldots, P_q$

② 열쇠 K

③ n비트인 시작변수 SV

[The variables employed for the CFB mode of operation are : a) The input variables ① A sequence of q plaintext variables $P_1, P_2, \ldots, P_q$, each of j bits. ② A key K. ③ A starting variable SV of n bits.]

b) 중간결과

① 각기 n비트인 q개의 알고리즘 입력 블럭열 $X_1, X_2, \ldots, X_q$.

② 각기 n비트인 q개의 알고리즘 출력 블럭열 $Y_1, Y_2, \ldots, Y_q$.

③ 각기 j비트인 q개의 변수열 $E_1, E_2, \ldots, E_q$.

④ 각기 k비트인 q-1개의 피트백 변수열 $F_1, F_2, \ldots, F_{q-1}$.

[b) The intermediate results ① A sequence of q algorithm input blocks $X_1, X_2, \ldots, X_q$, each of n bits. ② A sequence of q algorithm output block $Y_1, Y_2, \ldots, Y_q$ 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.]

c) 출력변수 즉, 각기 j비트인 q개의 암호문 변수열 $C_1, C_2, \ldots, C_q$. [The output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of j bits.]

6.2. 입력블럭 X의 초기값은 다음과 같이 지정한다.

$$X_1 = SV \quad \ldots (7)$$

각 평문 블럭을 암호화하는 운영은 다음 5개의 단계를 따른다.

a) 암호 알고리즘의 이용, $Y_i = eK(X_i)$ ... (8)

b) 왼쪽 j비트의 선택, $E_i = Y_i \sim j$ ... (9)

c) 암호문 변수의 생성, $C_i = P_i \oplus E_i$ ... (10)

d) 피드백 변수의 생성, $F_i = S_j(I(k)|C_i)$ ... (11)

e) 쉬프트 함수, $X_{i+1} = S_k(X_i|F_i)$ ... (12)

[The input block X is set to its initial value $X_1 = SV$ ... (7). The operation of enciphering each plaintext variable employs the following five steps: a) Use of encipherment algorithm, $Y_i = eK(X_i)$ ... (8), b) Selection of leftmost j bits, $E_i = Y_i \sim j$ ... (9), c) Generation of ciphertext variable, $C_i = P_i \oplus E_i$ ... (10), d) Generation of feedback variable, $F_i = S_j(I(k)|C_i)$ ... (11), e) Shift function, $X_{i+1} = S_k(X_i|F_i)$ ... (12)]

이 단계들은 i=1, 2, ..., q까지 반복되어, 마지막 사이클인 식(12)에서 끝난다. 이 절차를 그림 2의 왼쪽에 나타내었다. 암호화 알고리즘의 출력 블럭 Y의 왼쪽 j비트들은 j비트의 평문 변수를 2진수의 가산으로 암호화하는데 사용된다. Y의 나머지 비트

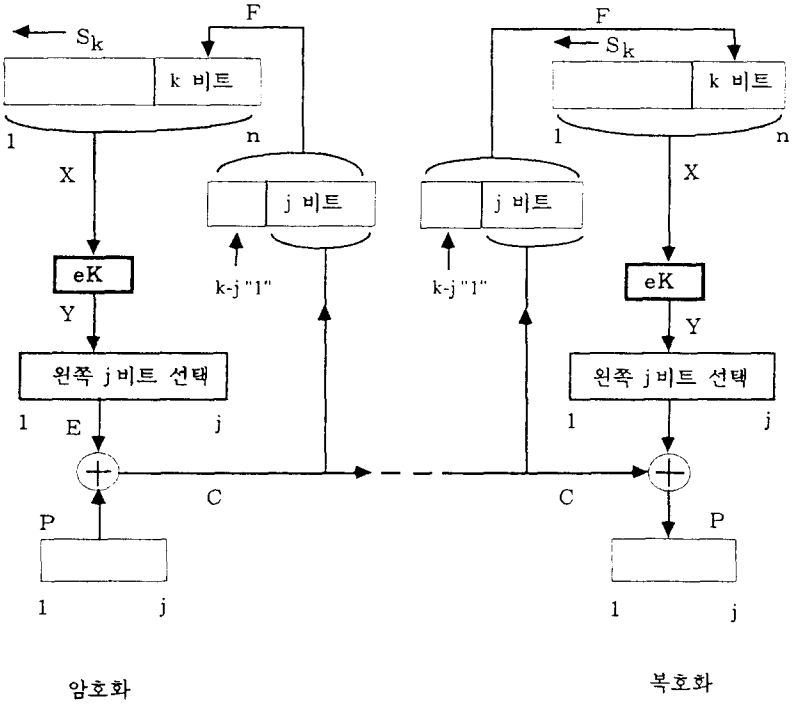


그림 2. 암호 피드백 (CFB) 운영 모드

는 생략된다. 평문과 암호문 변수들은 1에서 j까지 번호가 붙여진 비트를 갖는다. [These steps are repeated for i=1, 2, ..., q, ending with equation (12) 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 encipherment algorithm 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.]

암호문 변수는 왼쪽 비트 위치에 k-j개의 "1"을 대치하여 k비트 피드백 변수 F가 된다. 그때 입력블럭 X의 비트는 k만큼 왼쪽으로 쉬프트되고, F는 오른쪽에서 k비트만큼 삽입되어 새로운 X값을 생성한다. 이 쉬프트 연산에서는 X의 왼쪽 k비트가 생략된다. [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 input block X are shifted left by k places and F is inserted in the rightmost k places to produce the new value of X. In this shift operation, the leftmost k bits of X are discarded.]

6.3. 복호화에 사용된 변수들은 암호화에 사용된 것과 동일하다. 입력블럭 X는 초기값으로 X_1=SV가 지정된다. 각 암호문 블럭을 복호화하는 운영은 다음 5개의 단계들을 따른다.

a) 암호 알고리즘의 이용, $Y_i = eK(X_i)$ ... (13)
b) 왼쪽 j비트의 선택, $E_i = Y_i \sim j$ ... (14)
c) 평문 변수의 생성, $P_i = C_i \oplus E_i$ ... (15)
d) 피드백 변수의 생성, $F_i = S_j(I(k)|C_i)$ ... (16)
e) 쉬프트 함수, $X_{i+1} = S_k(X_i|F_i)$ ... (17)

[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 five steps: a) Use of encipherment algorithm, $Y_i = eK(X_i)$ ... (13) b) Selection of leftmost j bits, $E_i = Y_i \sim j$ ... (14) c) Generation of plaintext variable, $P_i = C_i \oplus E_i$ ... (15) d) Generation of feedback variable, $F_i = S_j(I(k)|C_i)$ ... (16) e) Shift function, $X_{i+1} = S_k(X_i|F_i)$ ... (17).]

이 단계들은 i=1, 2, ..., q까지 반복되어, 마지막 사이클인 식(17)에서 끝난다. 이 절차를 그림 2의 우측에 나타내었다. 암호 알고리즘의 출력블럭 Y의 왼쪽 j비트는 j비트의 암호문 블럭을 2진수의 가산으로 복호화하는데 사용된다. Y의 나머지 비트는 생략된다. 평문과 암호문 블럭은 1에서 j까지 번호가 붙여진 비트를 갖는다. [These steps are repeated for i=1, 2, ..., q, ending with equation (17) 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 encipherment algorithm 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.]

암호문 변수는 왼쪽 비트 위치에 k-j개의 "1"을 대치하여 k비트 피드백 변수 F가 된다. 그때 입력 블럭 X의 비트는 k만큼 왼쪽으로 쉬프트되고, F는 오른쪽에서 k비트만큼 삽입되어 새로운 X값을 생성한다. 이 쉬프트 연산에서는 X의 왼쪽 k비트가 생략된다. [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 input block X are shifted left by k places and F is inserted in the rightmost k places to produce the new value of X. In this shift operation, the left most k bits of X are discarded.]

6.4. CFB에서는 같은 j와 k값을 사용하도록 권고한다. 이 권고안 (j=k)에서 식(11)과 (16)은 다음과 같이 나타낸다. $F_i = C_i$ (j=k경우) [It is recommended that CFB should be used with equal values of j and k. In this recommended form (j=k) the

equations (11) and (16) can be written, $F_i=C_i$ (case j=k).]

7. 출력 피드백(OFB) 모드 [Output Feedback (OFB) Mode]

7.1. 한개의 파라미터가, 즉 평문 변수의 크기 j ($1 \le j \le n$), OFB 운영 모드를 정의한다. OFB 운영 모드에서 사용되는 변수들은 다음과 같다.

a) 입력변수
① 각기 j비트인 q개의 평문 변수열 $P_1, P_2, \ldots, P_q$
② 열쇠 K
③ n비트인 시작변수 SV

[One parameter defines an OFB mode of operation, i.e. the size of plaintext variable j where $1 \le j \le n$. The variables employed for the OFB mode of operation are: a) The input variables ① A sequence of q plaintext variables, $P_1, P_2, \ldots, P_q$, each of j bits. ② A key K. ③ A starting variable SV of n bits.]

b) 중간결과
① 각기 n비트인 q개의 입력 블럭열, $X_1, X_2, \ldots, X_q$.
② 각기 n비트인 q개의 출력 블럭열, $Y_1, Y_2, \ldots, Y_q$.
③ 각기 j비트인 q개의 변수열 $E_1, E_2, \ldots, E_q$.

[b) The intermediate results ① A sequence of q algorithm input blocks $X_1, X_2, \ldots, X_q$, each of n bits. ② A sequence of q algorithm output block $Y_1, Y_2, \ldots, Y_q$ each of n bits. ③ A sequence of q variables $E_1, E_2, \ldots, E_q$, each of j bits.]

c) 출력변수들은 각기 j비트인 q개의 암호문 변수열, $C_1, C_2, \ldots, C_q$. [The output variables, i.e. a sequence of q ciphertext variables $C_1, C_2, \ldots, C_q$, each of j bits.]

7.2. 입력블럭 X의 초기값은 다음과 같이 지정한다.

$$X_1=SV \quad \ldots(18)$$

a) 암호 알고리즘의 이용, $Y_i=eK(X_i)$ ...(19)
b) 왼쪽 j비트의 선택, $E_i=Y_i\sim j$ ...(20)
c) 암호문 변수의 생성, $C_i=P_i \oplus E_i$ ...(21)
d) 피드백 연산, $X_{i+1}=Y_i$ ...(22)

[The input block X is set to its initial value $X_1=SV$. The operation of enciphering each plaintext variable employs the following four steps: a) Use of encipherment algorithm, $Y_i=eK(X_i)$ ...(19), b) Selection of leftmost j bits, $E_i=Y_i\sim j$ ...(20), c) Generation of ciphertext variable, $C_i=P_i \oplus E_i$ ...(21), d) Feedback operation, $X_{i+1}=Y_i$ ...(22)]

이 단계들은 i=1, 2, ..., q까지 반복되며, 마지막 사이클인 식(21)에서 끝난다. 이 절차는 그림 3의 왼쪽에 그려져 있다. 암호 알고리즘을 사용한 매 번의 결과 Y_i는 피드백되고, X_i의 다음 값, 즉 X_{i+1}이 된다. Y_i의 좌측에서 j비트는 입력 변수를 암호화하는데 사용된다. [These steps are repeated for i=1,2, ..., q, ending with equation (21) on the last cycle. The procedure is shown in the left side of figure 3. The result of each use of the encipherment algorithm, which is Y_i is used to feedback and become the next value of X_i namely X_{i+1}. The leftmost j bits of Y_i are used to encipher the input variable.]

7.3. 복호화하는데 사용된 변수들은 암호화할 때 사용된 변수들과 같다. 입력블럭 X는 초기값으로 $X_1=SV$가 지정된다. 각 암호문 블럭을 복호화하는 운영은 다음 네 가지 단계들을 따른다.

a) 암호 알고리즘의 이용, $Y_i=eK(X_i)$ ...(23)
b) 왼쪽 j비트의 선택, $E_i=Y_i\sim j$ ...(24)
c) 평문 변수의 생성, $P_i=C_i \oplus E_i$ ...(25)
d) 피드백 운영, $X_{i+1}=Y_i$ ...(26)

[The variables employed for decipherment are the

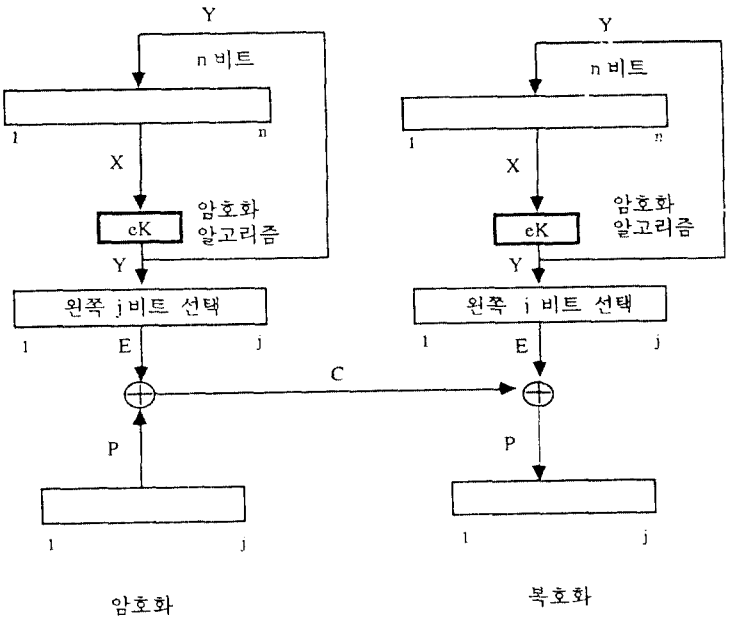


그림 3. 출력 피드백 (OFB) 운영 모드

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 encipherment algorithm, $Y_i = eK(X_i)$ ... (23) b) Selection of leftmost j bits, $E_i = Y_i \sim j$ ... (24) c) Generation of plaintext variable, $P_i = C_i \oplus E_i$ ... (25) d) Feedback operation, $X_{i+1} = Y_i$ ... (26).]

이 단계들은 i=1, 2, ... , q번까지 반복되며, 마지막 사이클인 식(25)에서 끝난다. 이 절차는 그림3의 오른쪽 부분에 그려져 있다. X_i와 Y_i의 값은 암호화에 사용되던 값과 같다 ; 단지 식(25)만이 다를 뿐이다. [These steps are repeated for i=1, 2, ..., q, ending with equation (25) 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 (25) is different.]

(참고) 부록 A : n비트 운영 모드의 특성
[Annex A (informative) : Properties of the modes of operation]

A.1. 전자 코드북(ECB) 운영 모드의 특성
[Properties of the Electronic Codebook (ECB) Mode of Operation]

A.1.1. 환경 [Environment]

컴퓨터 또는 인간 상호간의 정보를 운반하는 메세지는 반복성을 갖고있거나 자주 사용되는 열들로 되어 있다. ECB 모드에서 동일한 평문은(같은 열쇠일 경우) 동일한 암호문을 생성한다. [Messages that carry information between computers, or people, may have repetitions of commonly used sequences. In ECB mode, identical plaintext produces (for the same key) identical ciphertext blocks.]

A.1.2. 특성 [Properties]

ECB 모드의 특성은 다음과 같다.

a) 블럭의 암호화 또는 복호화는 다른 블럭과 독립적으로 처리가능하다.

b) 암호문 블럭의 재배열은 해당 평문 블럭의 재배열을 초래한다.

c) 동일 평문 블럭은 (같은 열쇠일 경우) 동일 암호문 블럭을 생성하기 때문에 "사전 공격"에 취약하다.

[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".]

ECB 모드는 일반적으로 한 블럭보다 긴 메세지에 대해서는 권장하지 않는다. 앞으로 새로운 국제표준에서는 ECB 모드의 반복적 성질과 블럭들이 독립적으로 접근되는 그러한 성질의 특별한 목적을 위한 범위에 대해서도 규정할 계획이다. [The ECB mode is in general not recommended for messages longer than one block. The use of ECB may be specified in future International Standard for those special purposes where the repetition characteristic is acceptable or blocks have to be accessed individually.]

A.1.3. 채워넣기 요구사항 [Padding requirements]

n비트의 정수배만이 암호화 및 복호화가 가능하다. 다른 길이는 n비트 경계까지 채워넣기가 필요하다. [Only multiples of n bits can be enciphered or deciphered. Other lengths need to be padded to a n-bit boundary.]

A.1.4. 오류의 전파 [Error propagation]

ECB 모드에서 단일 암호문 블럭에서 하나 혹은 그 이상의 오류는 복호문에서도 그 오류가 발생한 블럭 내에서만 영향을 미친다. 암호화를 함에 있어서 한개의 평문 비트의 변화는 암호문에 있어서 평균 50%의 변화를 초래한다는 가정하에, 이 블럭의 복호화된 평문은 평균 50%의 오류율을 가질 것이다. [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). Under the assumption that the cipher has the property that changing one plaintext bit results in an average 50% change in the ciphertext each bit of the recovered plaintext version of this block will have an average error rate of 50%.]

A.1.5. 블럭 경계 [Block boundaries]

암호화와 복호화 사이에서 블럭 경계를 잃어버리면 (즉 비트 슬립 때문에) 올바른 블럭 경계가 재성립될 때까지 암호화와 복호화 작업 사이에서의 동기화가 되지 않을 것이다. 모든 복호화 작업의 결과는 블럭 경계가 상실되어 있는 한 올바르게 되지 않을 것이다. [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. 암호 블럭 체이닝 (CBC) 운영 모드의 특성 [Properties of the Cipher Block Chaining (CBC) Mode of Operation]

A.2.1. 환경 [Environment]

CBC 모드에서는 같은 열쇠와 같은 시작 변수로 동일한 평문이 암호화되면 언제나 동일한 암호문이 생긴다. 이 특성에 대해서 걱정이되는 사용자는 평문의 시작, 열쇠 또는 시작 변수를 바꾸기 위한 어떤 방법을 택할 필요가 있다. 그러한 방법중 한가지는 각 CBC문의 시작 부분에 유일한 식별자(unique identifier), (즉 증가계수, incremented counter)를 포함시키는 것이다. 크기가 증가되어서는 안되는

레코드를 암호화할 때에 이용되는, 또 다른 방법으로는 그 내용(즉 랜덤 액세스 저장소에 들어있는 블럭 주소)을 알지 않고도 레코드에서 계산될 수 있는 시작 변수와 같은 어떤 값을 사용하는 방법이 있다. [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 (i.g. an incremented counter) at the beginning of each CBC message. Another, which may be used when 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. 특성 [Properties]

CBC 모드의 특성은 다음과 같다.

a) 체이닝 연산은 암호문 블럭들을 현재 그리고 선행되었던 모든 평문 블럭들에 연관성 있도록 만들므로, 블럭들은 재배열 될 수 없다.

b) 다른 SV값을 사용함으로써 동일한 평문이 동일한 암호문으로 암호화 되는 것을 방지한다.

[Properties of the CBC mode are: a) the chaining operation makes the ciphertext blocks dependent on the current and all proceeding plaintext blocks and therefore blocks can not be rearranged ; b) the use of different SV values prevents the same plaintext enciphering to the same ciphertext.]

A.2.3. 채워넣기 요구사항 [Padding requirements]

데이터 블럭이 n비트의 정수배일때만 암호화 또는 복호화가 가능하다. 다른 길이일때는 n비트 경계까지 채워넣기를 해주어야 한다. 만약 이런 방법이 허용되지 않으면, 마지막 변수는 특별한 방법으로써 처리될 수 있다. 특별한 방법에 대한 두 예제는 다음과 같다.

a) 암호화

$$C_q=P_q \oplus (eK(C_{q-1}\sim j) \quad \ldots (27)$$

b) 복호화

$$P_q=C_q \oplus (eK(C_{q-1}\sim j) \quad \ldots (28)$$

그러나, 만약 이 시작 변수가 공개되거나(비밀이 아니거나), 시작 변수가 동일한 열쇠로 한 번 이상 사용된다면, 마지막 변수는 "선택적 평문 공격"에 약한 단점이 있다.(A.4절 참조) [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) encipherment: $C_q=P_q \oplus (eK(C_{q-1}\sim j)$ ... (27). b) decipherment: $P_q=C_q \oplus (eK(C_{q-1}\sim j)$ ... (28). However, this last variable is vulnerable to a "chosen plaintext attack" if the SV is not secret or it it is used more than once with the same key.(see clause A.4)]

두번째 방법으로, "암호문-훔침"(ciphertext stealing)이 있다. 가령, 마지막 두 개의 변수가 P_{q-1}과 P_q일때, P_{q-1}은 n비트 블럭이고 P_q는 j비트(단, j<n 비트이고 q>1)라 하자. [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) 암호화

C_{q-1}이 5.2에서 기술된 방법으로 P_{q-1}에서 유도된 암호문 블럭이라 할때, C_q는 다음과 같이 된다.

$$C_q=eK(S_j(C_{q-1}|P_q) \quad \ldots (29)$$

마지막 두 개의 암호문은 $C_{q-1}\sim j$와 C_q이다.

b) 복호화

먼저 C_q를 복호화하고, 그 결과로 변수 P_q와 오른쪽으로 부터 n-j 비트의 C_{q-1}를 얻는다.

$$S_j(C_{q-1}|P_q)=dK(C_q) \quad \ldots (30)$$

이제 완전한 블럭 C_{q-1}는 알고, 5.3에서 기술된 방

법으로 P_{q-1}를 구할 수 있다. 마지막 두개의 암호문 변수는 복호화할때 순서가 뒤바뀌는 까닭에 하드웨어적으로 구현할때 이 방법이 해결책으로써 적당치 않다. [a) encipherment: Let C_{q-1} be the ciphertext block derived from P_{q-1} using the method described in 5.2. Then set $C_q=eK(S_j(C_{q-1}|P_q)$ ... (29). The last two ciphertext variables are than C_{q-1}~j and C_q. b) decipherment: C_q needs to be deciphered first, resulting in the variable P_q and the rightmost n-j bits of C_{q-1}, $S_j(C_{q-1}|P_q)=dK(C_q)$ ... (30). 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 va riables are deciphered in reverse order which makes this solution less suited for hardware implementations.]

A.2.4. 오류의 전파[Error propagation]

CBC 방식에서는 하나의 암호문 블럭내에 한개 또는 그 이상의 비트의 오류는 두 블럭(오류가 발생한 블럭과 그 다음의 블럭)의 복호화에 영향을 준다. 암호화에 있어서 한개의 평문 비트의 변화는 암호문에 있어서 50%의 변화를 초래한다는 가정하에서, 만약 i번째 암호문 블럭에 오류가 발생하면, i번째의 복호된 평문 블럭의 평균 비트 오류율은 50%이다. (i+1)번째 복호된 평문 블럭은 오류가 있는 암호문 비트에 직접적으로 일치하는 비트에만 오류가 발생하게 된다. 만약 n비트보다 적은 변수에서 오류가 발생하면 오류 전파는 선택된 특별한 처리 방식에 좌우된다. 첫번째 예는 복호화된 짧은 블럭은 오류가 있는 암호문 비트에 직접적으로 일치된 비트에만 오류가 발생한다. 만약 오류가 n비트보다 적은 블럭의 앞 블럭에서 발생한다면 복호화되는 짧은 블럭은 50%의 평균 비트 오류율을 보일 것이다. (두번째 예인) 암호문 훔침의 경우에는 짧은 블럭이나 마지막 암호문 블럭에서의 오류는 각각 50%의 비트 오류율을 나타낸다. [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). If the errors occur in the i th ciphertext block, each bit of the i th deciphered plaintext block will have an average error rate of 50%, under the assumption that the cipher has the property that changing one plaintext bit results in an average 50% change in the ciphertext. The (i+1) th deciphered plaintext block will have only those bits in error that correspond directly to the ciphertext bits in error. 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. If errors occur in a block preceding a block of less than n bits, the deciphered short block will show an average bit error rate of 50%. In the ciphertext stealing case errors in the short block or the last ciphertext block each result in a bit error rate of 50%.]

A.2.5. 블럭 경계[Block boundaries]

암호화와 복호화 사이에서 블럭 경계를 잃어버리면 (즉 비트 슬립 때문에) 올바른 블럭 경계가 재성립될 때까지 암호화와 복호화 운영 사이에서의 동기화가 되지 않을 것이다. 블럭 경계가 상실된 동안에는 모든 복호화 운영 결과는 정확하지 않을 것이다. [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. 암호 피드백(CFB) 운영 모드의 특성 [Properties of the Cipher Feedback (CFB) Mode of Operation]

A.3.1. 환경[Environment]

CFB 모드에서는 같은 열쇠와 같은 시작변수로

동일한 평문이 암호화되면 언제나 동일한 암호문이 생긴다. 이 특성에 대해서 걱정이 되는 사용자는 평문의 시작, 열쇠 또는 시작 변수를 바꾸기 위한 어떤 방법을 택할 필요가 있다. 한 가지 방법은 CFB 메시지의 시작 부분에 식별자(unique identifier), (즉 증가계수, incremented counter)를 포함시키는 것이다. 크기가 증가되어서는 안되는 레코드를 암호화할 때에 이용되는, 또 다른 방법으로는 그 내용(즉 랜덤 액세스 저장소에 들어있는 블럭 주소)을 알지 않고도 레코드에서 계산될 수 있는 시작 변수와 같은 어떤 값을 사용하는 방법이 있다. [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 content (e.g. its address in random access storage).]

A.3.2. 특성 [Properties]

a) 체이닝 연산은 암호문 변수를 현재 그리고 선행된 모든 평문 변수들에 연관성이 있도록 한다. 그러므로 j비트 변수들은 함께 체이닝되어 있으며 재배열될 수 없다.

b) 다른 SV값을 사용함으로써 동일한 평문이 동일한 암호문으로 암호화되는 것을 방지한다.

c) CFB 모드에서 암호화와 복호화 과정은 둘다 암호화 형태의 알고리즘을 사용한다.

d) CFB 모드의 힘은 K의 크기에 달려있다. (j=k 이면 최대)

e) 작은 j값을 선택하면 평문 단위당 암호 알고리즘의 싸이클이 많이 요구되어 프로세싱 오버헤드가 커진다.

[a) the chaining operation makes the ciphertext variables dependent on the current and all preceding plaintext variables and therefore j-bits variables are chained together and can not be rearranged ; 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 form of the algorithm ; d) the strength of the CFB mode depends on the size of k (maximal if j=k) ; e) selection of a small value of j will require more cycles through the encipherment algorithm per unit of plaintext and thus cause greater processing overheads.]

A.3.3. 채워넣기 요구사항 [Padding requirements]

j비트의 정수배만이 암호화 또는 복호화 될 수 있다. 다른 길이는 j-비트 경계까지 채워넣기를 할 필요가 있다. 그러나, 대부분의 응용에서는, j는 문자크기와 같게 선택되어지고 채워넣기가 요구되지 않는다. [Only multiples of j bits can be enciphered or deciphered. Other lengths need to be padded to a j-bit boundary. However, in most applications j will be chosen equal to the character size and no padding will be required.]

A.3.4. 오류의 전파 [Error propagation]

CFB 모드의 경우에는 j-비트 암호문에 어떤 오류가 있을때 CFB 입력 블럭에서 오류 비트들이 쉬프트되어 모두 빠져나갈 때까지 이어지는 암호문의 복호화에 영향을 미친다. 최초로 영향을 받은 j-비트 평문의 오류 비트들의 위치는 암호문에서의 오류 비트들의 위치와 같다. 암호화가 평문에서 한 비트를 바꿈으로써 암호문에서 50% 의 오류율을 가지는 성질이 있다는 가정하에 연속되는 복호화된 평문에서 각 비트는 모든 에러가 입력 블럭 밖으로 쉬프트되어질 때까지 평균 50% 의 오류율을 가질 것이다. [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 input block. The first affected j-bit unit of plaintext will be garbled in exactly those places where the ciphertext is in error. Under the assumption that the cipher has the property that changing one plaintext bit results in an average 50% change in the ciphertext, in the succeeding deciphered plaintext each bit will have an average error rate of 50% until all errors have been shifted out of the input block.]

A.3.5. 블럭 경계 [Block boundaries]

만약 j-비트 경계가 암호화와 복호화 사이에서 상실된다면 (즉 비트 슬립 때문에), 암호적 동기성은 j-비트 경계가 재성립된 후 다음의 n-비트에서 재성립될 것이다. 만약 j-비트의 정수배만큼 상실된다면 n비트 후에 동기화가 자동적으로 재성립될 것이다. [If j-bit boundaries are lost between encipherment and decipherment (e.g. due to a bit slip), cryptographic synchronization will be re-established n bits after j-bit boundaries are re-established. If a multiple of j bits are lost synchronization will be re-established automatically after n bits.]

A.4. 출력 피드백 (OFB) 운영 모드의 특성 [Properties of the Output Feedback (OFB) Mode of Operation]

A.4.1. 환경 [Environment]

출력 피드백 모드는 동일한 평문이 같은 열쇠와 시작 변수로 암호화될 경우 언제나 동일한 암호문을 생성한다. 게다가, 출력 피드백 모드에서는 같은 열쇠와 시작 변수가 사용될 경우 동일한 열쇠 스트림이 생성된다. 결과적으로, 보안상의 이유 때문에, 특정 시작변수는 주어진 열쇠에 대해 오직 한번만 사용되어야 한다. [The OFB mode produces the same ciphertext whenever the same plaintext is enciphered using the same key and starting variable. Moreover, in the OFB mode the same key stream 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.4.2. 특성 [Properties]

OFB 모드의 특성들은 다음과 같다.

a) OFB 모드는 체이닝이 없기 때문에 특정한 공격에 더욱 취약하다.

b) 다른 SV 값은 서로 다른 열쇠 스트림을 생성시키므로, 동일한 평문이 동일한 암호문으로 암호화되는 것을 방지해 준다.

c) OFB 모드에서는 암호화와 복호화 과정에서 둘다 같은 형태의 암호 알고리즘을 사용한다.

d) OFB 모드에서는 평문에 2진 덧셈 연산으로 하는데 쓰이는 열쇠 스트림 평문과는 무관하게 생성된다.

e) j의 값이 작아질수록, 평문의 매 단위당 암호화 알고리즘을 거치는 사이클의 수는 그만큼 많아지고, 더 많은 프로세싱 오버헤드가 발생한다.

[Properties of the OFB mode are: a) the absence of chaining makes the OFB more vulnerable to specific attacks ; b) the use of different SV values prevents the same plaintext enciphering to the same ciphertext, by producing different key streams ; c) the encipherment and decipherment processes in the OFB mode both use the encipherment form of the algorithm ; d) the OFB mode does not depend on the plaintext to generate the key stream used to add modulo 2 to the plaintext ; e) selection of a small value of j will require more cycles through the encipherment algorithm per unit of plaintext and thus cause greater processing overheads.]

A.4.3. 채워넣기 요구사항 [Padding requirements]

j비트의 정수배만이 암호화 또는 복호화가 가능하다. j비트의 정수배가 아닌 경우에는 j비트 경계까지 채워넣기를 할 필요가 있다. 그러나, 대부분의 응용에서, j는 문자 크기와 동일하게 선택되므로, 채워넣기가 필요없게 된다. [Only multiples of j bits can be enciphered or deciphered. Other lengths

need to be padded to a j-bit boundary. However, in most applications j will be chosen equal to the character size and no padding will be required.]

A.4.4. 오류의 전파[Error propagation]

OFB 모드는 암호문의 오류를 그 결과로 나오는 평문의 출력에 전파되지 않도록 한다. 암호문 상의 오류가 있는 각각의 비트는 복호화된 평문상에 오직 그 하나의 오류 비트만을 발생시킨다. [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 deciphered plaintext.]

A.4.5. 블럭 경계[Block boundaries]

OFB 모드는 자기동기화가 되지 않는다. 만약 암호화와 복호화 2개의 운영이 동기성을 잃는다면, 시스템은 재초기화를 할 필요가 있다. 이와같은 동기성의 상실은 j비트 블럭(j>1일때)의 경계 상실(즉 비트 슬립 때문에)에 의한 것일 수 있다. [The OFB mode is not self-synchronizing. If the two operations of encipherment and decipherment get out of synchronism, the system needs to be re-initialized. Such a loss of synchronism might be (if j>1) the loss of correct boundaries of the j-bit blocks (e.g. due to a bit slip).]

각각의 재초기화는 같은 열쇠를 가지고, 전에 사용했던 SV 값과는 다른 SV 값이 사용되어야 한다. 그 이유는 같은 파라미터들에 대해서는 매번 동일한 비트 스트림이 생성되기 때문이다. 이것은 "많은 평문과 암호문의 쌍으로부터 키를 찾는 공격"(known plaintext attack)에 대해 취약하다. [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 a "known plaintext attack".]

(참고) 부록 B : 특허에 관한 정보
[Annex B (informative) : Information about Patents]

이 국제표준을 준비하는 동안에, 이 국제표준의 응용에 관련이 될 만한 특허에 관한 정보들을 수집했다. IBM과 UNISYS에 속한 특허들을 명시했다. 그러나, ISO는 특허나 권리에 대한 증거, 사실여부, 혹은 범위에 대한 완전한 정보를 제공할 수 없다. [During the preparation of this International Standard, information was gathered concerning relevant patents upon which application of this International Standard might depend. Relevant patents were identified as belonging to International Business Machines Corporation (IBM) and UNISYS. However, ISO cannot give authoritative or comprehensive information about evidence, validity or scope of patent or like rights.]

특허-특허 소지자들은 이 국제표준이 응용 가능하게 하기위해 특허를 돌려준다는 전제하에 관련 특허를 인가해 주기로 했다. 더 상세한 정보는 다음 주소에서 가능합니다. [The patent-holders have stated that licenses will be granted in appropriate terms to enable application of this International Standard, provided that those who seek licenses agree to reciprocate. Further information is available from]

Director of Commercial Relations
International Business Machines Corporation
2000 Purchase Street
PURCHASE, N.Y. 10577
U.S.A.

Director, Industry Relations
UNISYS
P.O. Box 500
Blue Bell, PA 19424
U.S.A.

(참고) 부록 C : 운영 모드에 관한 보기
[Annex C (informative) : Examples for the Modes of Operation]

C.1. 개요 [General]

이 부록은 국제표준이 정한 운영모드를 사용한 메세지에 대한 암호화와 복호화에 관한 보기들이다. 이 보기들은 다음과 같은 파라미터들을 사용한다.

a) 사용된 암호화 알고리즘은 Data Encryption Algorithm(DEA). n의 값은 64이다.

b) 암호학적 열쇠는 0123456789ABCDEF.

c) 시작 변수는 1234567890ABCDEF.

d) 평문은 7비트 ASCII 코드에 해당하는 'Now is the time for all '(16진수 표기법 4E6F772069732074686520746965206 6F72616C6C20). CFB 모드에서 평문은 7비트 ASCII코드에 해당하는 'Now'(16진수 표기법 4E6F77).

[This annex gives examples for the encipherment and decipherment of a message using the modes of operation specified in this International Standard. The examples use the following parameters: a) The encipherment algorithm used is the Data Encryption Algorithm(DEA). The value of n is 64. b) The cryptographic key is 0123456789ABCDEF. c) The starting variable is 1234567890ABCDEF. d) The plaintext is the 7-bit ASCII code for 'Now is the time for all '(in hexadecimal notation 4E6F772069732074686520746965206 6F72616C6C20). For CFB mode the plaintext is the 7-bit ASCII code for 'Now'(in hexadecimal notation 4E6F77).]

C.2. ECB 모드

ECB 모드의 암호화와 복호화에 관한 보기들은 표 C.1과 C.2에 각각 있다. [Examples for the ECB mode of encipherment and decipherment are given in tables C.1 and C.2, respectively.]

C.3. CBC 모드

CBC 모드의 암호화와 복호화에 관한 보기들은 표 C.3과 C.4에 각각 있다. [Examples for the CBC

표 C.1 - ECB 모드, 암호화

i	평문, P_i	입력블럭	출력블럭	암호문, C_i
1	4E6F772069732074	4E6F772069732074	3FA40E8A984D4815	3FA40E8A984D4815
2	68652074696D6520	68652074696D6520	6A271787AB8883F9	6A271787AB8883F9
3	666F7220616C6C20	666F7220616C6C20	893D51EC4B563B53	893D51EC4B563B53

표 C.2 - ECB 모드, 복호화

i	암호문, C_i	입력블럭	출력블럭	평문, P_i
1	3FA40E8A984D4815	3FA40E8A984D4815	4E6F772069732074	4E6F772069732074
2	6A271787AB8883F9	6A271787AB8883F9	68652074696D6520	68652074696D6520
3	893D51EC4B563B53	893D51EC4B563B53	666F7220616C6C20	666F7220616C6C20

표 C.3 - CBC 모드, 암호화

i	평문, P_i	입력블럭	출력블럭	암호문, C_i
1	4E6F772069732074	5C5B2158F9D8ED9B	E5C7CDDE872BF27C	E5C7CDDE872BF27C
2	68652074696D6520	8DA2EDAAEE46975C	43E934008C389C0F	43E934008C389C0F
3	666F7220616C6C20	25864620ED54F02F	683788499A7C05F6	683788499A7C05F6

표 C.4 - CBC 모드, 복호화

i	암호문, C_i	입력블럭	출력블럭	평문, P_i
1	E5C7CDDE872BF27C	E5C7CDDE872BF27C	5C5B2158F9D8ED9B	4E6F772069732074
2	43E934008C389C0F	43E934008C389C0F	8DA2EDAAEE46975C	68652074696D6520
3	683788499A7C05F6	683788499A7C05F6	25864620ED54F02F	666F7220616C6C20

표 C.5 - CFB 모드, 암호화

i	평문, P_i	입력블럭	출력블럭	암호문, C_i
1	4E	1234567890ABCDEF	BD661569AE874E25	F3
2	6F	34567890ABCDEF*F3*	7039546F9A0F6330	1F
3	77	567890ABCDEFF3*1F*	AD1B78B0BB371BE7	DA

표 C.6 - CFB 모드, 복호화

i	암호문, C_i	입력블럭	출력블럭	평문, P_i
1	F3	1234567890ABCDEF	BD661569AE874E25	4E
2	1F	34567890ABCDEF*F3*	7039546F9A0F6330	6F
3	DA	567890ABCDEFF3*1F*	AD1B78B0BB371BE7	77

표 C.7 - OFB 모드, 암호화

i	평문, P_i	입력블럭	출력블럭	암호문, C_i
1	4E6F772069732074	1234567890ABCDEF	BD661569AE874E25	F3096249C7F46E51
2	68652074696D6520	BD661569AE874E25	5D976A504786581F	35F24A242EEB3D3F
3	666F7220616C6C20	5D976A504786581F	5B0229C3443694E3	3D6D5BE3255AF8C3

표 C.8 - OFB 모드, 복호화

i	암호문, C_i	입력블럭	출력블럭	평문, P_i
1	F3096249C7F46E51	1234567890ABCDEF	BD661569AE874E25	4E6F772069732074
2	35F24A242EEB3D3F	BD661569AE874E25	5D976A504786581F	68652074696D6520
3	3D6D5BE3255AF8C3	5D976A504786581F	5B0229C3443694E3	666F7220616C6C20

mode of encipherment and decipherment are given in table C.3 and C.4, respectively.]

C.4. CFB 모드

CFB 모드의 암호화와 복호화에 관한 보기들은 표 C.5와 C.6에 각각 있다. 이 보기에서는 파라미터 j=k=8이 선택되었다. 피드백 k비트는 이탤릭체로 표기되었다. [Examples for the CFB mode of encipherment and decipherment are given in tables C.5 and C.6, respectively. For this example the parameters j=k=8 have been chosen. The k bits feedback are shown in italic.]

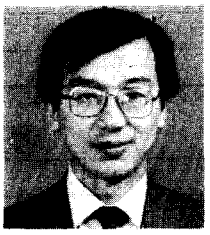
C.5. OFB 모드

OFB 모드의 암호화와 복호화에 관한 보기들은 표 C.6과 C.7에 각각 있다. 이 보기에서는 파라미터

j=64가 선택되었다. [Examples for the OFB mode of encipherment and decipherment are given in tables C.7 and C.8, respectively. For this example the parameter j=64 has been chosen.]

감사 : 이 표준의 번역을 도와준 정 호석군을 비롯한 포항공대 정보산업대학원 학생들에게 감사를 표한다.

□ 著者紹介



李 弼 中 (종신회원, 국제이사)

1951년 12월생
1974년 2월 서울대학교 전자공학과 학사
1977년 2월 서울대학교 전자공학과 석사
1982년 6월 U.C.L.A. System Science, Engineer
1985년 6월 U.C.L.A. Electrical Engineering, Ph.D.
1980년 6월~1985년 8월 : Jet Propulsion Laboratory, Senior Engineer
1985년 8월~1990년 2월 : Bell Communications Research, M.T.S
1990년 2월~현재 : 포항공과대학 전자전기공학과, 부교수.