● 대칭키 암호 운영모드

① **평문 길이**가 암호 알고리즘의 **블록 크기보다 클 때**는 어떻게 암호화 할 것인가?
→ 방법은, 평문을 암호 알고리즘의 블록 크기로 **분할**해서 암호화 하는 것이다.
② 긴 평문을 암호화하기 위한 여러 유형의 적절한 운영 방식이 있다.
→ 이런 운영 방식들을 "암호 알고리즘의 **운영모드**"라 한다.

// 운영모드

모드	설 명
ECB 모드	• Electric CodeBook mode(전자 부호표 모드) • 가장 **단순**한 모드이다. 평문을 n개의 블록으로 분할한다. • 평문 크기가 블록 크기의 배수가 아니면, 마지막 블록은 덧붙이기가 필요
CBC 모드	• Cipher Block Chaining mode(암호 블록 연쇄 모드) • 암호문 블록을 **체인**처럼 연결시키는 방식이다.
CFB 모드	• Cipher-FeedBack mode(암호 피드백 모드) • 이전 단계의 **암호문 블록**을 암호 알고리즘의 입력으로 사용한다.
OFB 모드	• Output-FeedBack mode(출력 피드백 모드) • 암호 알고리즘의 **출력**이 암호 알고리즘의 입력으로 사용된다.
CTR 모드	• CounTeR mode(카운터 모드) • 증가하는 **카운터**를 암호화해서 키 스트림을 생성하는 스트림 암호이다.

• ECB, CBC, CFB, OFB 모드는 DES에 적용할 목적으로 개발되었고
• CTR 모드는 2001년에 AES에 적용하기 위해 개발되었다.

// 암호 알고리즘의 수식 표현

암호화 : C = E_K(P)

복호화 : P = D_K(C)

P	평문(plaintext)
C	암호문(ciphertext)
E_K	비밀키 K를 적용한 암호 알고리즘
D_K	비밀키 K를 적용한 복호 알고리즘

운영모드 정리 - 한국정보보호진흥원 TTA저널 참조

모드	암호화, 복호화 수식 표현	특 징
ECB	암호화 : $C_i = E_K(P_i)$ 복호화 : $P_i = D_K(C_i)$	• 평문을 N개의 블록으로 분할 • 각 암복호화 하는데 **같은 키**를 사용한다.
CBC	암호화 : $C_i = E_K(P_i \oplus C_{i-1})$ 복호화 : $P_i = D_K(C_i) \oplus C_{i-1}$	• 평문 블록을 이전 암호문 블록과 **xor** 연산 • 그리고, 암호화를 실시한다.
CFB	암호화 : $C_i = P_i \oplus E_K(C_{i-1})$ 복호화 : $P_i = C_i \oplus E_K(C_{i-1})$	• 실제 암복호화를 위하여 **xor** 연산 사용 • 블록보다 작은 단위로 암호화가 가능하다.
OFB	암호화 : $C_i = P_i \oplus E_K(S_{i-1})$ 복호화 : $P_i = C_i \oplus E_K(S_{i-1})$	• CFB 모드와 매우 유사하다. • 암호문의 각 비트는 이전 암호문 블록과는 독립적
CTR	암호화 : $C_i = P_i \oplus E_K(N_i)$ 복호화 : $P_i = C_i \oplus E_K(N_i)$	• 카운터를 암호화한 후에 평문 블록과 **xor** 연산 • 카운터는 **초기 벡터값**으로 사용된다.

(주) P_i는 평문, C_i는 암호문, $E_K(P_i)$는 암호화, $D_K(C_i)$는 복호화를 의미한다.

• ECB, CBC, CTR은 **긴 메시지** 파일을 암호화 하는데 사용될 수 있고
• CFB, OFB는 **작은 데이터**를 암호화 하는데 사용될 수 있다.(Forouzan 참조. 240쪽)

• CFB, OFB, CTR은 복호화 연산 없이 **암호화 연산만 사용**된다.
• 즉, 복호화 과정에서 복호화 연산이 적용되지 않고 암호화 연산이 적용된다.

모드	유형(type)	초기 벡터	오류 전파	암호 병행처리	복호 병행처리
ECB	**블록 암호**	필요 없음(none)	No	가능	가능
CBC	**블록 암호**	필요함(yes)	**Yes**	**불가**	가능
CFB	**비동기식** 스트림 암호	필요함(yes)	**Yes**	**불가**	가능
OFB	동기식 스트림 암호	필요함(yes)	No	**불가**	**불가**
CTR	동기식 스트림 암호	필요함 - 카운터	No	가능	가능

• **오류 전파** 기준은 암호문의 특정 비트가 전송 도중에 **변조(손상, 오류)된 경우**이다.
• **오류 전파**는 암호문의 특정 비트가 **손실(누락)**된 기준이 아니다.

기출문제 분석

1. 블록암호 알고리즘의 운영모드로 옳지 않은 것은? [2015년 국가 7급]

① ECB(Electronic Codebook)
② CBC(Cipher Block Chaining)
③ CFB(Cipher Feedback)
④ ECC(Error Correction Code)

♤ **대칭키 암호 운영모드**

• ECB, CBC, CFB, OFB, CTR 모드 → ECC는 없다.

정답 : ④

2. 대칭키 블록 암호 알고리즘의 운영모드 중에서 한 평문 블록의 오류가 다른 평문 블록의 암호 결과에 영향을 미치는 오류 전이(error propagation)가 발생하지 않는 모드만을 묶은 것은?(단, ECB: Electronic Code Book, CBC: Cipher Block Chaining, CFB: Cipher Feedback, OFB: Output Feedback) [2018년 국가 9급]

① CFB, OFB　② ECB, OFB
③ CBC, CFB　④ ECB, CBC

♤ **대칭키 암호 운영모드**

• 오류 전이(error propagation)가 발생하지 않는 모드는 ECB, OFB, CTR이다.

모드	유형(type)	초기 벡터(initialization vector)	오류 전파
ECB	**블록 암호**	필요 없음(none)	No
CBC	**블록 암호**	필요함(yes)	**Yes**
CFB	스트림 암호	필요함(yes)	**Yes**
OFB	스트림 암호	필요함(yes)	No
CTR	스트림 암호	필요함(yes) – 카운터	No

• 오류 전파 기준은 암호문의 특정 비트가 전송 도중에 변조(손상, 오류)된 경우이다.
• 오류 전파는 암호문의 특정 비트가 손실(누락)된 기준이 아니다.

정답 : ②

3. 블록암호 알고리즘 운영모드에 대한 설명으로 가장 옳지 않은 것은? [2020년 서울 7급]

① CBC 모드는 현재의 평문 블록을 암호키로 암호화한 후 바로 직전의 암호블록과 XOR 연산을 수행한다.

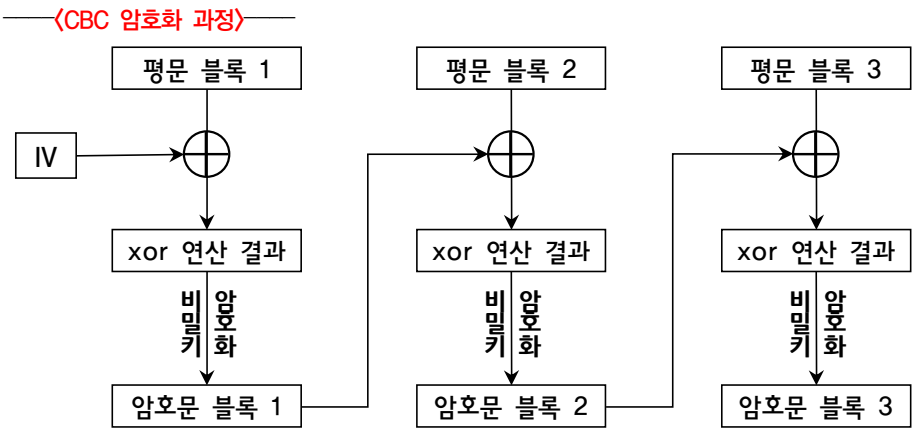
② CFB 모드 동작에서는 평문 블록 내에 한 비트의 오류가 발생하면 모든 암호문에 영향을 미치게 된다.

③ OFB 모드는 전송 중에 비트 오류가 전파되지 않는 동기식 스트림 암호이다.

④ CTR 모드는 패딩이 필요 없고 암호화 및 복호화의 사전 준비를 할 수 있어 병렬처리가 가능하다.

♤ **운영모드**

• CBC는 평문 블록과 이전 단계의 암호문 블록을 XOR 연산 후, 암호화를 수행한다.



// 운영모드 정리

모드	유형(type)	초기 벡터	오류 전파	암호 병행처리	복호 병행처리
ECB	**블록 암호**	필요 없음(none)	No	가능	가능
CBC	**블록 암호**	필요함(yes)	**Yes**	**불가**	가능
CFB	비동기식 스트림 암호	필요함(yes)	**Yes**	**불가**	가능
OFB	동기식 스트림 암호	필요함(yes)	No	**불가**	**불가**
CTR	동기식 스트림 암호	필요함 - 카운터	No	가능	가능

• 오류 전파 기준은 암호문의 특정 비트가 전송 도중에 변조(손상, 오류)된 경우이다.

• 오류 전파는 암호문의 특정 비트가 손실(누락)된 기준이 아니다.

정답 : ①

4. 다음은 블록 암호의 운영모드 중 하나를 표현하고 있다. 해당 운영모드에 대해 추론할 수 있는 설명으로 옳은 것은? (단, $i, j \geq 0$, $i \neq j$이다) [2022년 국가 7급]

P_i: 평문 블록	$C_i = P_i \oplus O_i$
C_i: 암호문 블록	$P_i = C_i \oplus O_i$
E_k: 암호화 함수(키 k 이용)	$O_i = E_K(I_i)$
IV: 초기벡터(initial vector)	$I_i = O_{i-1}$(단, $I_0 = IV$)

① C_i에 비트 오류가 발생하더라도 복호화된 P_j에 영향을 미치지 않는다.
② P_i와 P_j가 동일할 경우 C_i와 C_j가 같아지는 문제점이 존재한다.
③ 고속의 암복호화를 위해 별도의 전처리 없이 병렬처리가 가능하다.
④ 복호화에 사용되는 IV값은 암호화에 사용된 IV값과 다를 수 있다.

♤ **운영모드**

// 먼저, 암복호화 과정을 분석하면 다음과 같다.

i = 0	• $C_0 = P_0 \oplus O_0 = P_0 \oplus E_K(I_0) = P_0 \oplus E_K(IV)$ • $P_0 = C_0 \oplus O_0 = C_0 \oplus E_K(I_0) = C_0 \oplus E_K(IV)$
i = 1	• $C_1 = P_1 \oplus O_1 = P_1 \oplus E_K(I_1) = P_1 \oplus E_K(O_0) = P_1 \oplus E_K(E_K(I_0)) = P_1 \oplus E_K(E_K(IV))$ • $P_1 = C_1 \oplus O_1 = C_1 \oplus E_K(I_1) = C_1 \oplus E_K(O_0) = C_1 \oplus E_K(E_K(I_0)) = C_1 \oplus E_K(E_K(IV))$

• i=0이면 IV를 1번 암호한 것과 암복호화 한다.
• i=1이면 IV를 2번 암호한 것과 암복호화 한다.
• i=2이면 IV를 3번 암호한 것과 암복호화 한다.
• i=i이면 IV를 i+1번 암호한 것과 암복호화 한다.

① C_i에 비트 오류가 발생하더라도 복호화된 P_j에 영향을 미치지 않는다.(○)
→ C_i와 P_j는 서로 아무런 **연관성 없이 암복호화 처리**되므로($i \neq j$)
② P_i와 P_j가 동일할 경우 C_i와 C_j가 **같아지는 문제점이 존재**한다.(×)
→ **같아지는 문제점이 존재**하지 않는다. $i \neq j$이므로
③ 고속의 암복호화를 위해 별도의 전처리 없이 **병렬처리**가 **가능**하다.(×)
→ **병렬처리**가 불**가능**하다. 이전에 IV를 암호 처리한 것이 필요하므로
④ 복호화에 사용되는 IV값은 암호화에 사용된 IV값과 **다를 수 있다.**(×)
→ IV값과 **같아야 한다.** 다르면 정상적으로 암복호화가 될 수 없다.

정답 : ①