Q1. 블록 길이가 8바이트인 암호알고리즘을 이용하여 7바이트 메시지를 CBC 모드, PKCS #7로 암호화하였다고 가정하자. 이때 결과 암호문이 16진수로 “07 06 05 04 03 02 01 01 08 09 0A 0B 0C 0D 0E 0F”라고. 가정 하자. 또한 암호화한 데이터는 모르지만 7바이트이므로 16진수 01이 padding되었을 것이다. 만약 공격자가 IV 중 마지막 두 바이트 “01 01”을 “00 02”로 바꾸었지만 수신자가 복호화하는 과정에서 문제점을 발견하지 못하였다. 그러면 원래 메시지의 7번째 바이트 값을 계산하시오.

A1. 공격자가 IV 중 마지막 두 바이트를 01 01 대신 00 02로 바꾸어도 수신자는 복호화 과정에서 오류를 발견하지 못합니다. 그 이유는 바로 마지막 블록에서 padding 바이트의 값이 02가 되어야 하는데, 이미 평문이 01 01 01 01 01 01 01 01로 채워져 있기 때문입니다. 이렇게 되면 복호화한 결과의 마지막 블록은 81 0F 0F 0E 0F 0F 0E 0E XOR (IV': 07 06 05 04 03 02 00 02) = 88 09 0D 0D 08 0D 0D 0F입니다.
따라서 원래 메시지의 7번째 바이트는 0D입니다.

Q2. 블록 길이가 4바이트인 암호알고리즘을 이용하여 16진수 “00 00 01 01”을 암호화한 결과가 “01 01 01 01 05 06 07 08”이라고 가정하자. 여기서 “01 01 01 01”은 IV이다. 그러면 16진수 “01 01 01 01”을 암호화한 유효한 암호문을 제시하라.

A2. 블록 길이가 4바이트이므로 16진수 4자리는 한 블록을 의미합니다. CBC 모드에서는 이전 블록의 암호문과 현재 블록의 평문을 XOR한 값을 암호화합니다. 첫 번째 블록은 IV를 이용합니다.
주어진 암호문으로부터 첫 번째 블록의 평문을 알 수 있습니다. 암호문에서 첫 번째 블록은 "01 01 01 01"을 XOR한 값인 "01 01 01 01"입니다.
두 번째 블록의 평문은 암호문에서 "05 06 07 08"을 이전 블록의 암호문 "01 01 01 01"과 XOR한 값입니다. 즉, "04 07 06 09"입니다.
따라서 IV가 "01 01 01 01"일 때 평문이 "01 01 01 01"인 메시지를 암호화한 유효한 암호문은 "01 01 01 01 04 07 06 09"입니다.

Q3. CBC 모드를 암호화할 때 어떤 IV를 사용할지 예측이 된다고 가정하자. 또한 특정 메시지를 보내 암호화를 요청할 수 있는 공격자가 있다고 가정하자. 이 경우 공격자는 암호문 C0, C1이 평문 M을 암호화한 것인지 확인할 수 있다. 그 방법을 설명하시오.

A3. 공격자는 M1을 바꾸고 싶은 값으로 변경한 뒤, 이전 블록의 암호문과 XOR하여 새로운 C1'을 생성합니다. 이렇게 생성한 C1'을 피해자에게 전송하여, 피해자는 이전 블록의 암호문과 XOR하여 M1'을 얻습니다. 따라서 공격자는 메시지를 변경할 수는 없지만, 블록 단위로 메시지를 변경하여 암호문을 변조할 수 있습니다.

A4. CBC 모드를 이용하면 최종 암호문 블록을 MAC 값으로 충분히 활용할 수 있다고 하지만 실제 마지막 블록을
그대로 MAC으로 활용하면 공격자가 쉽게 위조할 수 있다. 공격자는 이 방식의 MAC을 위조하기 위해 한 블록 크기의 메시지 M에 대한 CBC-MAC 값 t을 얻으면 공격자는 t가 (M||t ⊕

M)에 대한 MAC임을 주장할 수 있다. 그 이유를 설명하시오. CBC-MAC 값을 계산할 때 IV는 모든 비트가 0인 블록을 사용한다.

A4. CBC-MAC은 CBC 모드를 이용하여 메시지의 블록들을 암호화하고, 마지막 블록을 MAC 값으로 사용하는 방식입니다. 즉, 메시지의 블록들을 암호화하는 과정에서 사용한 IV와 암호화된 마지막 블록이 MAC 값으로 사용됩니다.

공격자가 한 블록 크기의 메시지 M에 대한 CBC-MAC 값을 t로 얻었다고 가정하겠습니다. 그리고 공격자는 다음과 같은 메시지를 생성합니다: (M || t ⊕ M). 이 메시지를 CBC-MAC으로 암호화하면 다음과 같이 나타낼 수 있습니다.

- 첫 블록: IV ⊕ M
- 두 번째 블록: (IV ⊕ M) ⊕ t ⊕ M = IV ⊕ t

이 때, 두 번째 블록이 공격자가 얻은 CBC-MAC 값 t와 일치합니다. 따라서 공격자는 t가 (M || t ⊕ M)에 대한 MAC 값임을 주장할 수 있습니다.

이러한 공격을 방지하기 위해서는 MAC 값을 계산할 때, 마지막 블록 대신 다른 블록들의 함수값을 활용하거나, 마지막 블록을 별도로 암호화하여 MAC 값을 계산해야 합니다. 또한, IV 값도 메시지와 무관하게 랜덤하게 생성하여 사용해야 합니다.

Q5.디스크 단위 암호화(디스크 전체를 하나의 키로 투명하게 암호화함)에서 고려해야 하는 특성을 나열하고, 디스크 단위 암호를 위해 ECB, CBC, CTR 모드를 사용하는 것의 문제점을 설명하시오.

A5. 디스크 단위 암호화는 디스크 전체를 하나의 키로 투명하게 암호화하는 기술로, 주로 데이터 보호와 안전한 데이터 이전 등의 목적으로 사용된다. 이를 위해 고려해야 하는 특성은 다음과 같다.

1. 고속성: 대부분의 디스크 단위 암호화 기술은 대용량 디스크에 적용되므로, 암호화 및 복호화 작업이 빠르고 효율적이어야 한다.

2. 안정성: 암호화 키와 암호화 방법이 안전하며, 디스크 데이터가 누설되더라도 데이터가 유출되지 않도록 보호되어야 한다.

3. 호환성: 암호화된 디스크는 다른 시스템에서도 사용할 수 있도록 호환성이 있어야 한다.

ECB, CBC, CTR 모드는 디스크 단위 암호화에서 사용될 수 있지만, 각각의 모드에는 다음과 같은 문제점이 있다.

1. ECB 모드: 같은 평문 블록에 대해서는 항상 같은 암호문 블록이 생성되므로 패턴을 파악

하여 암호문에서 패턴을 찾아내어 공격할 수 있다.

2. CBC 모드: 복호화 시 이전 암호문 블록이 현재 블록의 복호화에 사용되므로, 암호문 블록이 손실되거나 변경되면 이후 블록의 복호화도 영향을 받는다. 또한, 암호화된 데이터의 첫 블록에 대해서는 이전 블록이 없으므로 다른 방법으로 MAC 값을 계산해야 한다.

3. CTR 모드: 같은 IV를 사용하면 평문 블록과 XOR하는 값들이 중첩될 수 있어서 중첩되면 두 평문 블록을 XOR한 값을 얻어낼 수 있다. 또한, IV 중 일부를 랜덤하게 생성하는 방식은 중첩 가능성이 낮아지지만, 랜덤하게 생성하는 방식보다 보안성이 떨어진다.