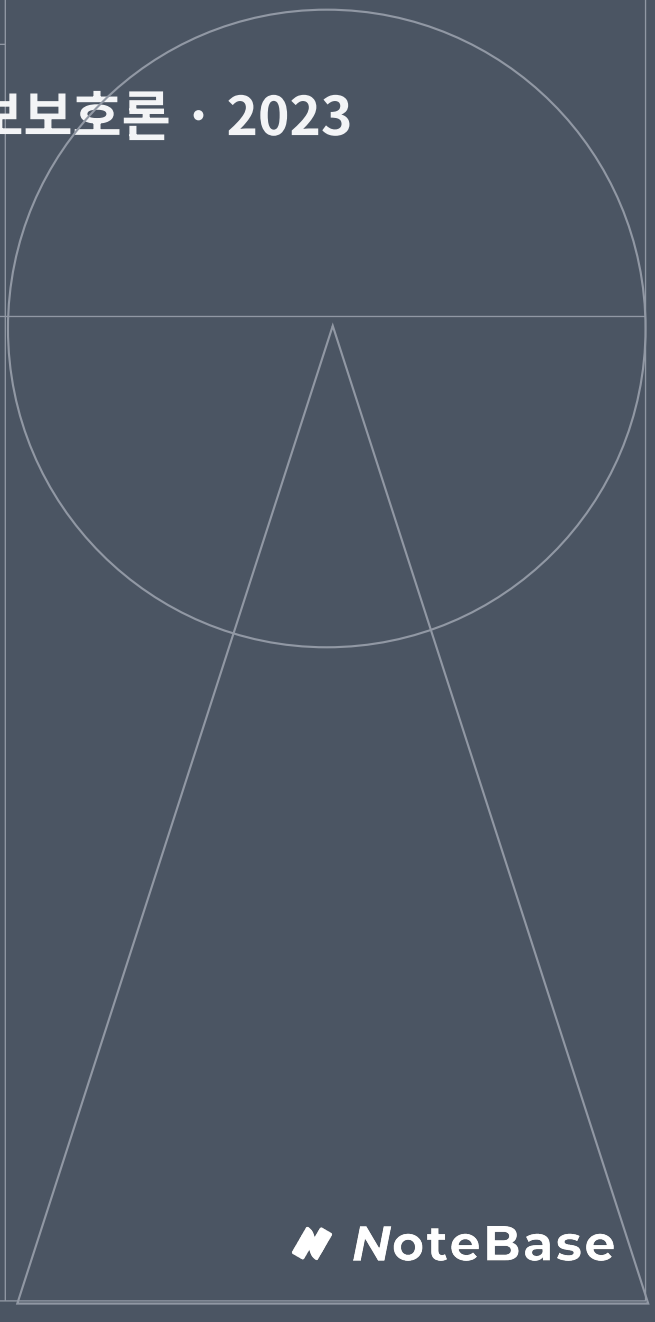
# 기출OX 해설집

공무원/경찰/경력채용 · 정보보호론 · 2023



NoteBase

# 기출OX 해설집

## 공무원/경찰/경력채용 · 정보보호론 · 2023

### 공무원/경찰/경력채용 · 정보보호론 · 2023

**문 1. OSI 7 계층 모델의 네트워크 계층에서 제공하는 보안 서비스로 가장 적절한 것은?**

① PPTP(Point-to-Point Tunneling Protocol)

② SSL(Secure Socket Layer)

③ IPSec(IP Security Protocol)

④ L2TP(Layer 2 Tunneling Protocol)

**정답 3번**

**간단한 해설**

IPsec은 **네트워크 계층의 IP 패킷**에 인증·무결성·기밀성 등의 보안 기능을 제공하는 프로토콜 체계이다. PPP 프레임을 터널링하는 PPTP·L2TP, TCP 위에서 동작하는 SSL과 적용 위치가 다르다.

**개념 구분**

OSI 모델의 네트워크 계층은 제3계층이며 IP가 여기에 해당한다. 이 문제는 터널을 운반하는 하위 프로토콜의 이름보다 **어느 계층의 데이터에 보안 기능을 적용하는지**를 구분한다.

IPsec: **IP 계층에서 패킷을 보호**하는 보안 프로토콜 체계이다. 사용하는 프로토콜과 설정에 따라 데이터 무결성, 송신 출처 인증, 재전송 공격 방지, 기밀성 등의 서비스를 제공한다. 상위 응용프로그램에 공통으로 적용할 수 있는 네트워크 계층 보안 방식이다.

PPTP와 L2TP: **PPP 프레임을 터널링**하는 제2계층 터널링 방식이다. PPTP는 PPP 패킷을 GRE로 캡슐화하여 IP망으로 운반하고, L2TP도 PPP 프레임을 캡슐화하여 전달한다. IP망이나 UDP를 운반 수단으로 사용한다는 사실만으로 IP 계층 자체의 보안 프로토콜이 되는 것은 아니다.

SSL: **TCP 같은 신뢰성 있는 전송 프로토콜 위**에서 동작하며, 그 위에서 이용하는 응용프로토콜의 통신을 보호한다. 따라서 IP 계층에서 패킷을 보호하는 IPsec과 구별된다.

따라서 네트워크 계층의 보안 서비스를 묻는 조건에 해당하는 것은 ③ IPSec이다.

**문 2. 스트림 암호에 관한 설명으로 가장 적절하지 않은 것은?**

① 한 번에 1비트 혹은 1바이트의 데이터 흐름을 순차적으로 처리하는 암호 알고리즘을 총칭한다.

② 스트림 암호는 하드웨어 구현이 간편하고 속도가 빠르기 때문에 무선 통신 환경에 주로 사용된다.

③ 평문과 비밀키를 XOR 연산하여 암호화한다.

④ 대표적인 방식으로 RC5, IDEA 등이 있다.

**정답 4번**

**간단한 해설**

RC5와 IDEA는 일정한 크기의 데이터 블록을 변환하는 **블록 암호**이다. 스트림 암호의 대표 방식으로 분류할 수 없다.

**선지별 해설**

① O: 스트림 암호는 평문을 비트나 바이트 등의 작은 단위로 순차 처리하는 방식이다. 블록 암호가 정해진 길이의 평문 블록을 변환하는 것과 달리, 생성된 키스트림을 데이터 흐름에 차례로 결합한다. 다만 내부 구현에서 여러 비트를 묶어 처리할 수 있으므로, 모든 연산이 반드시 1비트 또는 1바이트씩만 이루어진다는 뜻은 아니다.

② O: 간결한 회로와 빠른 연속 처리가 가능한 스트림 암호는 지연과 구현 비용을 줄여야 하는 무선 통신에 적합하다. 이동통신에 쓰이는 SNOW 3G가 그 예이다. 이는 대표적인 활용 특성이며, 구현의 간편성·속도는 알고리즘과 장치에 따라 달라지고 무선 통신에서만 사용하는 것도 아니다.

③ X: 비밀키 자체를 평문과 그대로 XOR한다는 표현은 정확하지 않다. 통상적인 스트림 암호는 비밀키와 초기화 값 등을 이용해 키스트림을 만들고, **평문과 키스트림을 XOR**하여 암호문을 얻는다. 선지는 키스트림 생성 과정을 생략한 표현이므로, 고정된 비밀키를 직접 반복 XOR하는 방식과 혼동하지 않아야 한다.

④ X: RC5와 IDEA는 모두 **블록 암호**이다. RC5는 정해진 크기의 입력 블록을 같은 크기의 출력 블록으로 변환하고, IDEA는 64비트 블록을 암호화한다. 스트림 암호의 예로는 ChaCha20과 SNOW 3G를 들 수 있다. ③의 표현상 부정확성과 별개로, ④는 블록 암호를 스트림 암호로 분류한 직접적인 오류이므로 이 문항의 인정 정답에 해당한다.

NoteBase

**문 3. OSI 보안 구조 권고안 X.800에서 정의하는 적극적 공격(active attack)에 관한 설명으로 가장 적절하지 않은 것은?**

① 서비스 거부 공격을 한다.

② 인증된 사용자로 신분을 위장한다.

③ 전송 메시지를 수정한다.

④ 네트워크 트래픽을 분석한다.

**정답 4번**

**간단한 해설**

트래픽 분석은 통신을 변경하지 않고 그 양상에서 정보를 얻는 **수동적 공격**이다. 적극적 공격은 메시지나 시스템의 동작에 개입한다.

**선지별 해설**

적극적 공격은 정보의 내용이나 시스템의 상태·동작에 변화를 일으키는 공격이다. 수동적 공격은 이러한 변경 없이 정보를 관찰하여 알아내는 공격이다.

① O: 서비스 거부 공격은 정상 사용자가 서비스를 이용하지 못하도록 동작을 방해하여 가용성을 침해한다. 메시지를 차단하거나 과도한 트래픽을 발생시키는 등 **정상적인 서비스 동작에 개입**하므로 적극적 공격이다.

② O: 다른 사용자인 것처럼 행세하는 것은 위장(masquerade) 공격이다. 인증된 사용자의 신분을 사칭하여 허가받지 않은 접근이나 동작을 시도하므로 적극적 공격에 해당한다.

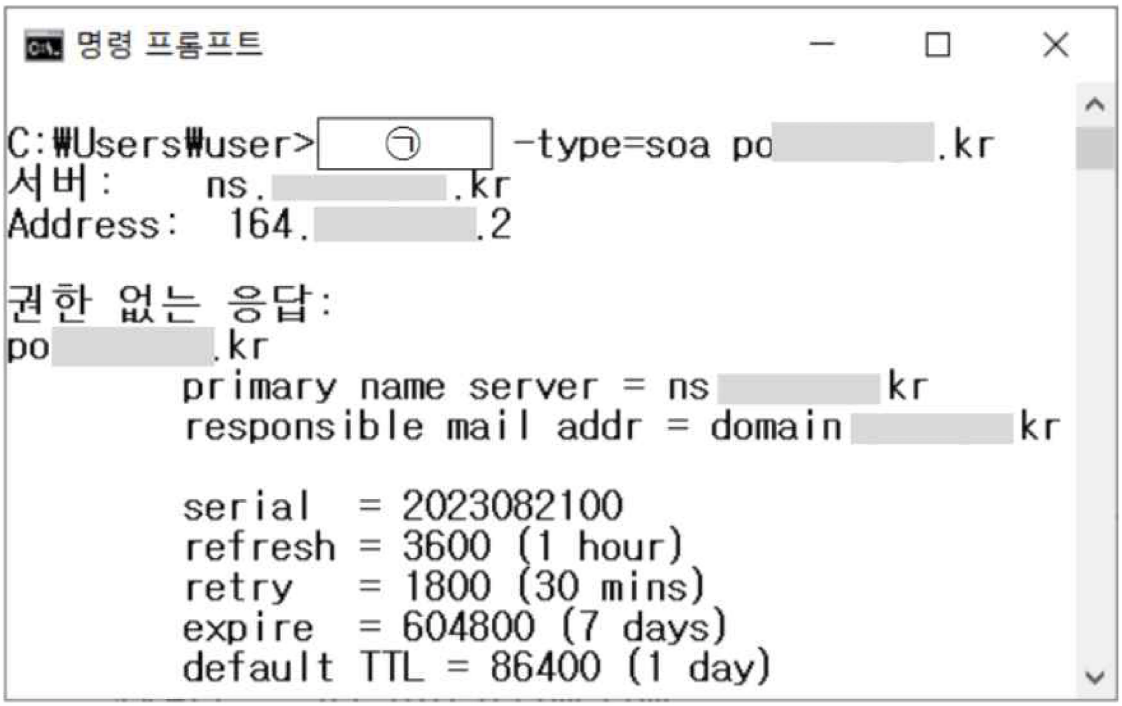
③ O: 전송 메시지를 수정하는 것은 데이터의 내용을 무단으로 변경하는 공격이다. 관찰에 그치지 않고 **메시지의 무결성을 침해**하므로 적극적 공격이다.

④ X: 네트워크 트래픽의 양·발생 시점·송수신 관계 등을 관찰하여 통신 양상을 추론하는 트래픽 분석은 **수동적 공격**으로 분류한다. 메시지 내용이 암호화되어 있어도 이러한 통신 특성을 분석할 수 있으며, 적극적 공격으로 분류한 부분을 고쳐야 한다.

따라서 적극적 공격에 관한 설명으로 가장 적절하지 않은 것은 ④이다.

NoteBase

**문 4. 다음 그림은 DNS 진단 유틸리티 명령어를 실행한 결과의 일부이다. ㉠의 명령어로 옳은 것은?**




① route

② ipconfig

③ ping

④ nslookup

**정답 4번**

**간단한 해설**

**-type=soa**는 DNS의 SOA 레코드를 조회하도록 지정하는 옵션이다. 영역의 주 이름 서버·관리자·일련번호와 관련 시간 값이 출력되는 화면은 nslookup의 실행 결과이다.

NoteBase

**핵심 풀이**

명령어 판별: **nslookup**은 DNS 서버에 질의하여 이름 해석과 DNS 레코드를 확인하는 진단 도구이다. 그림의 -type=soa는 조회할 레코드를 SOA(Start of Authority)로 지정한 옵션이므로 명령 형식은 nslookup -type=soa 도메인명이다. 따라서 ㉠은 ④ nslookup이다.

출력의 단서: 위쪽의 서버와 Address는 질의에 사용한 DNS 서버 정보이다. 이어서 primary name server, responsible mail addr, serial, refresh, retry, expire, default TTL이 표시되어 있으므로 단순한 IP 주소 조회가 아니라 DNS 영역의 SOA 정보 조회 결과임을 알 수 있다. 그림에 있는 '권한 없는 응답' 역시 DNS 응답의 성격을 표시한 문구이며, 이를 명령 실행 자체가 실패했다는 뜻으로 해석하지 않는다.

① route와 구별: route는 로컬 IP 라우팅 테이블을 조회하거나 수정하는 명령이다. 경로 정보가 아니라 SOA 레코드가 출력된 이 화면과 다르다.

② ipconfig와 구별: ipconfig는 자신의 IP 주소·서브넷 마스크·기본 게이트웨이 등 TCP/IP 설정을 확인하는 명령이다. DNS 캐시를 조회하거나 비우는 옵션도 있지만, -type=soa로 원격 DNS 레코드를 질의하는 명령은 아니다.

③ ping과 구별: ping은 ICMP Echo 요청과 응답으로 상대 호스트의 IP 수준 연결 상태와 왕복 시간을 확인한다. 호스트 이름을 사용할 수 있더라도 SOA 레코드의 항목들을 출력하는 진단 도구는 아니다.

NoteBase

**문 5. TCP 프로토콜을 사용하는 서비스명과 일반적인 포트 (well-known port) 번호가 바르게 연결된 것은?**

① FTP: 21, SMTP: 25, HTTP: 80, Telnet: 23

② FTP: 23, SMTP: 25, HTTP: 80, Telnet: 21

③ FTP: 21, SMTP: 80, HTTP: 25, Telnet: 23

④ FTP: 80, SMTP: 23, HTTP: 21, Telnet: 25

**정답 1번**

**간단한 해설**

FTP의 제어 연결은 **21번**, SMTP는 **25번**, HTTP는 **80번**, Telnet은 **23번**을 기본 TCP 포트로 사용한다. 각 서비스의 포트를 이 순서대로 연결하면 된다.

**대응 관계 해설**

이 문제는 TCP를 사용하는 각 서비스의 기본 서버 포트를 연결하는 문제이다. 포트 번호는 같은 호스트에서 통신을 처리할 서비스나 프로그램을 구분한다.

FTP: **21번**은 파일 전송 명령과 응답을 주고받는 제어 연결의 기본 포트이다. 파일 데이터 연결과 제어 연결은 구별되므로, FTP의 서비스 포트를 묻는 이 문항에서는 21번을 대응시킨다.

SMTP: **25번**은 전자우편을 전송하는 SMTP의 기본 포트로, 특히 메일 서버 사이의 전달에 사용된다. 웹 문서를 주고받는 HTTP의 80번과 혼동하지 않는다.

HTTP: **80번**은 웹 자원을 요청하고 응답받는 HTTP의 기본 포트이다. TLS로 보호하는 HTTPS의 기본 포트인 443번과 구별된다.

Telnet: **23번**은 원격 호스트에 접속하여 터미널 방식으로 명령을 실행하는 Telnet의 기본 포트이다. FTP 제어 연결의 21번과 서로 바꾸면 안 된다.

따라서 FTP 21, SMTP 25, HTTP 80, Telnet 23의 대응을 제시한 ①이 정답이다.

NoteBase

**문 6. 다음 사례에 적용된 정보보호 기법으로 가장 적절한 것은?**

Alice는 우연히 오래된 지도를 발견하였다. 그 지도를 불빛에 비추어 보니, 보물이 숨겨져 있는 위치가 지도에 표시되었다.

① 스테가노그래피(steganography)
② 시저(caesar) 암호
③ 스키테일(scytale) 암호
④ 플레이페어(playfair) 암호

**정답 1번**

**간단한 해설**

지도 속 보물의 위치는 평소에 보이지 않다가 불빛에 비추었을 때 드러난다. 이처럼 매체 안에 **정보의 존재 자체를 숨기는** 방식은 스테가노그래피이다.

**핵심 풀이**

사례의 핵심은 지도에 있던 보물의 위치가 평소에는 드러나지 않다가 **불빛에 비추었을 때 나타났다는 점**이다. 글자나 기호를 다른 것으로 바꾸거나 배열을 다시 맞추어 뜻을 해독한 상황은 제시되지 않았다.

스테가노그래피: 전달하려는 정보를 다른 매체에 감추어 **비밀 정보가 있다는 사실 자체를 알아차리기 어렵게** 하는 기법이다. 사례에서는 지도가 숨겨진 위치 정보를 담는 매체이고, 불빛에 비추는 행동이 그 정보를 드러내는 조건이다. 구체적으로 어떤 잉크나 재료를 썼는지는 주어지지 않았으므로 단정할 수 없다.

시저 암호: 알파벳을 일정한 간격만큼 이동시켜 다른 문자로 바꾸는 **치환 암호**이다. 일정한 문자 대응으로 내용을 변환하며, 정보의 존재를 지도 안에 감추는 사례와 다르다.

스키테일 암호: 띠를 원통에 감아 적은 문자가 띠를 풀면 다른 순서로 읽히도록 하는 **전치 암호**이다. 같은 굵기의 원통에 다시 감아 배열을 복원하는 것이 핵심이며, 빛을 비추어 숨겨진 표시를 발견하는 원리와 다르다.

플레이페어 암호: 문자 두 개를 한 쌍으로 묶어 정해진 문자표와 규칙에 따라 다른 문자쌍으로 바꾸는 치환 암호이다. 숨겨진 정보의 존재를 드러내는 대신 **문자쌍의 내용을 변환**한다.

따라서 이 사례의 정보보호 기법은 ① 스테가노그래피이다.

NoteBase

**문 7. 클라우드 컴퓨팅에 관한 설명으로 가장 적절하지 않은 것은?**

① 컨테이너(containers) 기술을 이용하면 호스트 운영체제를 공유하기 때문에 각기 다른 게스트 운영체제를 가지는 가상머신에 비해 오버헤드가 적은 이점이 있다.

② 클라우드 구현방식에 따른 배치 모델(deployment model) 중에 프라이빗 클라우드(private cloud) 모델은 특정 기관이 내부적으로 구축하여 이용하는 모형이다.

③ 클라우드 환경은 온프레미스(on-premises) 환경에 비해 많은 초기 구매 비용이 소요되는 단점이 있다.

④ 가상화를 위한 하이퍼바이저(hypervisor)는 각종 하드웨어 자원을 각각의 가상머신에 논리적으로 할당 및 스케줄링하는 기능을 담당한다.

**정답 3번**

**간단한 해설**

서비스형 클라우드는 필요한 컴퓨팅 자원을 제공받아 사용하므로 자체 장비 구축에 비해 **초기 구매 비용을 줄일 수 있다**. 초기 구매 비용이 더 많이 든다는 설명은 이러한 비용 구조를 거꾸로 설명한 것이다.

**선지별 해설**

① O: 일반적인 운영체제 수준 컨테이너는 **호스트 운영체제의 커널을 공유**하면서 응용프로그램의 실행 환경을 격리한다. 가상머신마다 별도의 게스트 운영체제와 커널을 구동하는 방식에 비해 운영체제 중복 실행에 따른 메모리 사용과 시작 비용 등이 작다.

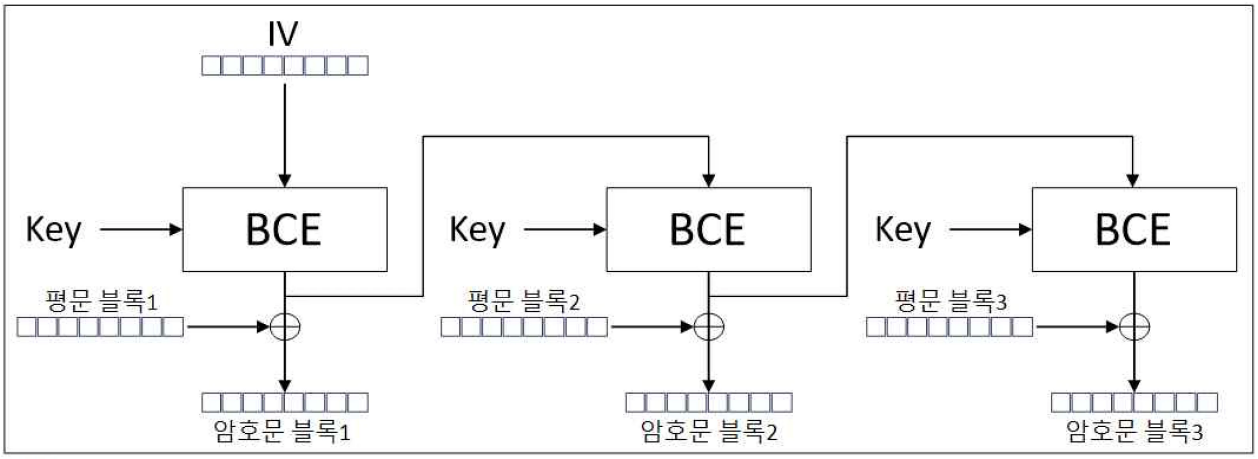
② O: 프라이빗 클라우드는 **하나의 조직이 전용으로 이용**하는 클라우드이며, 선지처럼 기관 내부에 구축해 사용하는 형태가 대표적인 예이다. 다만 내부 구축이 필수 요건은 아니다. 제3자가 운영하거나 외부 시설에 설치하더라도 한 조직 전용으로 제공되면 프라이빗 클라우드가 될 수 있다.

③ X: 일반적인 서비스형 클라우드 이용은 서버·저장장치 등을 직접 대량 구매하지 않고 필요한 자원을 제공받으므로, 온프레미스 구축보다 **초기 설비 구매 부담을 줄일 수 있다**. 온프레미스는 기관이 자체 시설에 장비를 설치·운영해 초기 투자가 커지기 쉽다. 다만 자체 구축형 프라이빗 클라우드에는 초기 투자가 필요하며, 클라우드의 장기 총비용이 언제나 더 낮다는 뜻도 아니다.

④ O: 하이퍼바이저는 물리적 CPU·메모리·입출력 장치 등의 자원을 가상화하여 여러 가상머신에 제공하고, 각 가상머신의 자원 접근을 조정한다. 예를 들어 여러 가상 CPU가 물리 CPU에서 실행될 시간을 배분하는 스케줄링을 수행해 자원을 공유하도록 한다.

NoteBase

**문 8. 다음 그림은 특정한 블록 암호 운영 모드에 관한 암호화 과정이다. 키 스트림 영향으로 병렬처리가 불가능하고 암호문 블록 오류 발생 시 해당 평문 블록에만 영향을 주어 오류 영향의 전파를 최소화한 모드로 가장 적절한 것은?**




** BCE : Block Cipher Encryption | IV : Initialization Vector

① ECB(Electronic CodeBook) 모드

② CBC(Cipher Block Chaining) 모드

③ CFB(Cipher FeedBack) 모드

④ OFB(Output FeedBack) 모드

**정답 4번**

**간단한 해설**

그림에서는 **블록 암호화의 출력 자체를 다음 입력으로 피드백**하므로 OFB 모드이다. 키 스트림은 순차적으로 생성되지만 암호문 비트 오류는 다음 평문 블록으로 전파되지 않는다.

NoteBase

**도식 해설**

그림의 피드백 위치: IV를 첫 번째 BCE(블록 암호화 연산)에 넣고, 그 출력은 평문 블록과 XOR하여 암호문을 만든다. 동시에 **XOR 이전의 BCE 출력**이 다음 BCE의 입력으로 되돌아간다. 암호문 자체를 되먹임하는 구조가 아니므로 출력 피드백인 OFB이다.

키 스트림의 순차 생성: 키를 $K$, 초기값을 $O_0 = IV$라 하면 출력 블록은 $O_j = E_K(O_{j-1})$, 암호문은 $C_j = P_j \oplus O_j$로 구한다. 여기서 $P_j$는 평문 블록이다. 다음 출력 블록을 만들려면 이전 출력이 필요하므로 BCE 연산들을 독립적으로 병렬 수행할 수 없다. 다만 키 스트림은 평문과 무관하므로 IV를 알면 미리 순차 생성할 수 있다.

오류 영향: 복호화는 $P_j = C_j \oplus O_j$이다. 암호문의 한 비트가 뒤집히면 같은 위치의 평문 비트만 뒤집힌다. 수신 암호문이 키 스트림 생성에 입력되지 않으므로 **다음 평문 블록으로 오류가 전파되지 않는다**. 이는 암호문 비트 오류의 영향에 관한 성질이지, 오류를 탐지하거나 무결성을 보장한다는 뜻은 아니다.

다른 모드와의 구별: ECB는 평문 블록을 각각 독립적으로 암호화하여 그림과 같은 피드백이 없다. CBC는 평문과 이전 암호문을 XOR한 뒤 암호화한다. CFB는 블록 암호화 출력과 평문을 XOR해 얻은 암호문을 다음 단계에 피드백하므로, BCE 출력 자체를 피드백하는 그림과 다르다.

따라서 그림과 설명에 해당하는 것은 ④ OFB 모드이다.

NoteBase

**문 9. RSA(Rivest–Shamir–Adleman) 암호 알고리즘에서 선택된 두 소수 ($p$, $q$)는 각각 7과 17이며 암호화 공개키($e$)가 5로 선택된 경우, 공개키와 개인키가 바르게 짝지어진 것은?**

① 공개키={e, N}: PU={5, 119}, 개인키={d, N}: PR={77, 119}

② 공개키={e, N}: PU={77, 119}, 개인키={d, N}: PR={5, 119}

③ 공개키={e, N}: PU={77, 5}, 개인키={d, N}: PR={5, 77}

④ 공개키={e, N}: PU={5, 17}, 개인키={d, N}: PR={5, 119}

**정답 1번**

**간단한 해설**

두 소수의 곱으로 공통 모듈러스 119를 구하고, **곱셈 역원**으로 개인 지수를 구한다. $\varphi(N) = 96$에 대해 5×77≡1±od96이므로 공개 지수는 5, 개인 지수는 77이다.

**계산 과정**

공통 모듈러스: 두 소수를 곱하면 $N = pq = 7 \times 17 = 119$이다. 공개키와 개인키에는 모두 이 값이 들어간다.

역원 계산의 기준: $\varphi(N) = (p-1)(q-1) = 6 \times 16 = 96$이다. 주어진 공개 지수는 $e = 5$이며, gcd(5, 96) = 1이므로 96을 법으로 하는 곱셈 역원이 존재한다.

개인 지수 계산: 기본 RSA 키 생성식 ed≡1±od(N)에 대입하면 5d≡1±od96이다. $96 = 19 \times 5 + 1$에서 $1 = 96 - 19 \times 5$를 얻으므로 d≡-19≡77±od96이다. 실제로 $5 \times 77 = 385 = 4 \times 96 + 1$이므로 $d = 77$은 조건을 만족한다.

키 구성과 선지 연결: 문제의 표기 순서에 따라 공개키는 $PU = \{e, N\} = \{5, 119\}$, 개인키는 $PR = \{d, N\} = \{77, 119\}$이다. 따라서 ①이 정답이다. ②는 주어진 공개 지수 5와 개인 지수의 자리를 바꾸었고, ③·④는 두 키가 공통으로 사용하는 모듈러스 119를 올바르게 배치하지 않았다.

NoteBase

**문 10. SHA(Secure Hash Algorithm)에 관한 설명으로 가장 적절하지 않은 것은?**

① SHA-1과 SHA-2의 해시 길이는 모두 다르다.

② SHA-1과 SHA-256의 블록 길이는 동일하다.

③ SHA-384와 SHA-512의 워드 길이는 동일하다.

④ SHA-1과 SHA-512의 단계 수는 서로 다르다.

**정답 4번**

**간단한 해설**

SHA-1과 SHA-512의 단계 수는 모두 **80단계**로 같다. 해시 길이·블록 길이·워드 길이가 서로 다를 수 있어도 단계 수까지 달라지는 것은 아니므로 ④가 적절하지 않다.

**선지별 해설**

해시 길이는 최종 출력의 비트 수, 블록 길이는 메시지를 나누어 처리하는 단위, 워드 길이는 내부 연산 단위이다. 단계 수는 한 블록을 처리할 때 반복하는 압축 연산의 횟수이므로 서로 구별해야 한다.

① O: SHA-1의 해시 길이는 **160비트**이다. SHA-2 계열의 SHA-224·SHA-256·SHA-384·SHA-512는 각각 224·256·384·512비트를 출력하므로 어느 것도 SHA-1과 같지 않다. SHA-512/224와 SHA-512/256의 출력도 각각 224·256비트이므로 이 비교는 달라지지 않는다.

② O: SHA-1과 SHA-256은 모두 메시지를 **512비트 블록**으로 나누어 처리한다. 최종 해시 길이는 각각 160비트와 256비트로 다르지만, 입력을 처리하는 블록 길이는 같다.

③ O: SHA-384와 SHA-512는 모두 **64비트 워드**를 사용한다. 두 알고리즘은 1024비트 블록을 64비트 단위로 처리하며, 최종 출력 길이가 각각 384비트와 512비트라는 차이가 워드 길이의 차이를 뜻하지는 않는다.

④ X: SHA-1과 SHA-512는 한 블록을 처리할 때 모두 **80단계**의 압축 연산을 수행한다. 따라서 단계 수가 서로 다르다는 부분을 같다고 고쳐야 한다. SHA-256이 64단계를 수행한다는 사실과 혼동하기 쉽다.

NoteBase

**문 11. 패스워드 암호화 강도를 높이기 위해 사용하는 첨가값(salt)에 관한 설명으로 가장 적절하지 않은 것은?**

① 사전(dictionary) 공격과 레인보우 테이블(rainbow table) 공격을 효과적으로 막을 수 있다.

② 첨가값을 사용하면 전사적(brute-force) 공격에 안전하다.

③ 의사난수생성기로 첨가값을 생성한다.

④ 비밀번호와 분리하여 안전한 장소에 저장한다.

**정답 2번**

**간단한 해설**

솔트는 미리 계산한 해시의 재사용을 어렵게 하지만 **전사적 공격 자체를 막지는 못한다**. 공격자는 알려진 솔트를 넣어 후보 비밀번호를 계속 검사할 수 있으므로, 솔트 사용만으로 안전을 보장할 수 없다.

**선지별 해설**

솔트(salt)는 비밀번호를 해시할 때 함께 넣는 사용자별 첨가값이다. 같은 비밀번호라도 서로 다른 솔트를 사용하면 저장되는 해시가 달라져, 공격자가 미리 만든 계산 결과를 여러 계정에 재사용하기 어려워진다.

① O: 충분히 길고 사용자별로 다른 솔트는 **미리 계산한 해시 사전과 레인보우 테이블의 재사용**을 어렵게 한다. 선지의 방어 효과는 이러한 사전 계산 공격에 대한 효과로 이해해야 한다. 솔트를 알아낸 공격자가 사전의 후보 비밀번호마다 해당 솔트를 넣어 새로 계산하는 추측 공격까지 없어지는 것은 아니다.

② X: 솔트를 사용해도 **전사적 공격에 대한 안전이 보장되지는 않는다**. 공격자가 솔트와 저장된 해시를 알면 후보 비밀번호를 하나씩 대입하여 결과를 비교할 수 있다. 솔트는 계정별로 계산하게 만들어 공격 비용을 높이지만, 약한 비밀번호를 강한 비밀번호로 바꾸지는 않는다. 충분히 강한 비밀번호와 계산 비용이 큰 비밀번호 전용 해시 방식 등을 함께 사용해야 한다.

③ O: 의사난수생성기로 솔트를 생성할 수 있다. 보안 용도에 적합한 생성기를 사용하여 **사용자별로 서로 다른 값**이 생성되도록 하고, 길이를 충분히 확보해 우연한 중복 가능성을 낮추는 것이 중요하다.

④ O: 솔트를 비밀번호와 구분해 안전하게 관리하는 방식은 가능하다. 다만 이 문구를 일반 솔트가 반드시 비밀이어야 하거나 비밀번호 해시와 다른 장소에 있어야 한다는 뜻으로 받아들이면 부정확하다. 일반 솔트는 **비밀일 필요가 없으며 결과 해시와 함께 저장할 수 있다**. 평문 비밀번호를 저장하는 방식은 피해야 한다. 해시 저장소와 분리하여 비밀로 보관해야 하는 추가 비밀값인 페퍼(pepper)와 일반 솔트를 구별한다.

①의 사전 계산 방어 효과와 ④의 관리 방식에 관한 한계를 고려하더라도, 솔트 사용만으로 전사적 공격에 안전하다고 보장하는 ②가 발문에서 묻는 가장 적절하지 않은 설명이다.

NoteBase

**문 12. 하이브리드 암호시스템에 관한 설명으로 가장 적절하지 않은 것은?**

① 대칭키의 키 배송 문제를 공개키 암호로 해결하였다.

② 의사난수생성기, 대칭키 암호, 공개키 암호 기술이 사용된다.

③ 공개키 암호로 속도를 높이고 대칭키 암호로 세션키를 보호한다.

④ 대표적인 사용 예로 PGP(Pretty Good Privacy)가 있다.

**정답 3번**

**간단한 해설**

하이브리드 암호는 **대칭키 암호로 메시지를 빠르게 처리**하고 **공개키 암호로 세션키를 보호**한다. 선지는 두 암호 방식의 역할을 반대로 설명했다.

**선지별 해설**

① O: 메시지를 암호화할 대칭 세션키를 수신자의 공개키로 암호화해 보내면, 수신자는 자신의 개인키로 세션키를 복원할 수 있다. 대칭키를 미리 안전하게 전달해야 하는 부담을 공개키 암호로 해결하는 방식이다. 이때 사용하는 공개키가 실제 수신자의 것인지 확인하는 절차는 필요하다.

② O: 암호학적으로 안전한 의사난수생성기는 예측하기 어려운 세션키를 생성하고, 대칭키 암호는 메시지 본문을 암호화하며, 공개키 암호는 세션키를 보호하는 데 쓰인다. 세 기술이 각각 **키 생성·데이터 암호화·키 전달**의 역할을 나누어 수행한다.

③ X: 대칭키 암호와 공개키 암호의 역할을 서로 바꾸어 설명했다. 대용량 메시지는 상대적으로 처리 속도가 빠른 **대칭키 암호로 암호화**하고, 그때 사용한 **세션키는 공개키 암호로 보호**하여 전달한다. 이렇게 두 방식의 장점을 결합한 것이 하이브리드 암호시스템이다.

④ O: PGP는 전자우편 등의 기밀성을 제공할 때 메시지를 대칭 세션키로 암호화하고, 세션키는 수신자의 공개키로 암호화하여 함께 전달하는 하이브리드 방식을 사용한다. 수신자는 개인키로 세션키를 복원한 뒤 그 세션키로 메시지를 복호화한다.

NoteBase

**문 13. 다음은 보안 알고리즘의 기능별 지원 여부(O, X)를 나타낸 표이다. ㉠~㉣에 해당하는 것으로 가장 옳게 짝지어진 것은?**

| 알고리즘 | 암호/복호 | 디지털서명 | 키교환 |
|---|---|---|---|
| SEED | O | ㉠ | X |
| DH(Diffie-Hellman) | ㉡ | X | O |
| ECC(Elliptic Curve Cryptography) | O | ㉢ | O |
| DSS(Digital Signature Standard) | X | O | ㉣ |

① ㉠(X) ㉡(X) ㉢(O) ㉣(O)

② ㉠(O) ㉡(O) ㉢(X) ㉣(O)

③ ㉠(X) ㉡(X) ㉢(O) ㉣(X)

④ ㉠(X) ㉡(O) ㉢(X) ㉣(O)

**정답 3번**

**간단한 해설**

DH의 **키합의**와 DSS의 **디지털서명**은 서로 다른 기능이다. SEED는 대칭키 암호이므로 디지털서명을 직접 제공하지 않으며, ECC 기반의 ECDSA는 디지털서명을 지원한다.

NoteBase

**빈칸별 해설**

표의 O와 X는 해당 기능의 지원 여부를 뜻한다. 암호·복호는 데이터의 내용을 숨기고 복원하는 기능이며, 디지털서명은 서명자와 데이터의 무결성을 확인하는 기능이다. 키교환은 통신에 사용할 공통 비밀값이나 키를 마련하는 절차이다.

㉠ X: SEED는 같은 비밀키를 사용하는 대칭키 블록 암호로, 암호화와 복호화를 수행한다. 개인키로 서명하고 공개키로 검증하는 **디지털서명 알고리즘은 아니다**. 표에서 SEED의 암호·복호는 O이고 키교환은 X이다.

㉡ X: DH는 양쪽이 각자의 비밀값과 상대방의 공개값을 이용해 공통 비밀값을 계산하는 **키합의 방식**이다. 이렇게 얻은 값으로 암호키를 마련할 수 있지만, DH 자체가 메시지를 암호화하거나 복호화하는 것은 아니다. 디지털서명도 직접 수행하지 않으므로 표의 키교환만 O이다.

㉢ O: ECC는 타원곡선에 기초한 공개키 암호 기술의 계열이며, ECDSA와 같은 **디지털서명 방식**을 구성할 수 있다. 또한 ECIES와 같은 암호화 방식, ECDH와 같은 키합의 방식에도 쓰인다. 표의 세 기능이 O인 것은 ECC 기반의 서로 다른 방식으로 각 기능을 지원한다는 뜻이다.

㉣ X: DSS는 **디지털서명 생성·검증을 위한 표준**이다. 서명을 통해 데이터 변조 여부와 서명자를 확인하는 것이 목적이며, 통신용 공통 비밀키를 합의하는 키교환 표준은 아니다. 따라서 표의 디지털서명은 O이고 암호·복호와 키교환은 X이다.

㉠ X, ㉡ X, ㉢ O, ㉣ X의 조합이므로 ③과 일치한다.

NoteBase

**문 15. 다음은 리눅스에서 ls -l 명령어를 실행한 결과이다. 파일의 접근권한을 소유자 외에는 읽기 권한만 부여하는 명령어로 옳은 것은?**

-rwxrwxrwx 1 AAA BBB 120 Oct 2 10:10 CCC

① chmod 666 AAA

② chmod 666 CCC

③ chmod 744 BBB

④ chmod 744 CCC

**정답 4번**

**간단한 해설**

변경할 파일은 **CCC**이다. 소유자에게 rwx, 그룹과 기타 사용자에게 읽기만 허용하는 권한은 **744**이므로 chmod 744 CCC를 선택한다.

**핵심 풀이**

변경 대상: ls -l 출력에서 AAA는 소유자, BBB는 소유 그룹, **CCC는 파일명**이다. chmod는 지정한 파일의 접근 권한을 바꾸는 명령이므로 마지막 인수에는 CCC를 넣어야 한다. 맨 앞의 -는 일반 파일임을 나타낸다.

권한 구성: rwxrwxrwx는 소유자·그룹·기타 사용자 순으로 읽기(r), 쓰기(w), 실행(x)이 모두 허용된 상태이다. 소유자의 rwx는 유지하고 **그룹과 기타 사용자를 r--로 변경**하면 rwxr--r--가 된다.

숫자 표현: 각 사용자 구분에서 읽기는 4, 쓰기는 2, 실행은 1로 표시하고 허용할 값을 더한다. 소유자의 rwx는 4 + 2 + 1 = 7, 그룹과 기타 사용자의 r--는 각각 4이므로 권한값은 744이다. 666은 모든 사용자 구분에 읽기와 쓰기를 허용하므로 읽기만 허용하라는 조건에 맞지 않는다.

명령어 선택: 권한값 744와 대상 파일 CCC를 결합한 **chmod 744 CCC**, 즉 ④가 정답이다. AAA나 BBB를 명령의 마지막에 쓰면 표시된 소유자나 그룹을 지정하는 것이 아니라 그 이름의 파일을 대상으로 삼게 된다.

NoteBase

**문 16. 다음은 통합인증체계(SSO, Single Sign On)를 위한 커버로스(kerberos) 프로토콜에 관한 내용이다. ㉠~㉢에 들어갈 용어가 올바르게 짝지어진 것은?**

실제 인증을 수행하는 인증 서버는 KDC(Key Distribution Center)의 부분 서비스로 각 서버와 유일한 (㉠)를 공유한다. 그리고 (㉡)는 인증 서버로부터 인증받은 사용자에게 티켓을 발행하는 서비스를 수행한다.
커버로스에서는 다른 사람이 티켓을 복사하여 실제 사용자로 위장하는 등의 재전송 공격을 막기 위해 발행된 티켓에 시간제한을 두는 (㉢)를 이용한다.

① ㉠ 비밀키 ㉡ AS(Authentication Server) ㉢ 쿠키

② ㉠ 비밀키 ㉡ TGS(Ticket Granting Server) ㉢ 타임스탬프

③ ㉠ 공개키 ㉡ TGS(Ticket Granting Server) ㉢ 쿠키

④ ㉠ 공개키 ㉡ AS(Authentication Server) ㉢ 타임스탬프

**정답 2번**

**간단한 해설**

커버로스는 **공유 비밀키**를 바탕으로 인증하며, 초기 인증 후에는 TGS가 서비스 티켓을 발급한다. 시간에 따른 유효성 판단에는 타임스탬프를 이용한다.

**빈칸별 해설**

㉠ 비밀키: 커버로스의 기본 인증 구조는 **서로 공유하는 대칭키**를 사용한다. KDC는 각 서버와 공유한 장기 비밀키를 이용해 그 서버용 티켓을 암호화하므로 해당 서버가 티켓을 복호화할 수 있다. 여기서 묻는 공유 키는 공개키가 아니다.

㉡ TGS(Ticket Granting Server): AS는 초기 인증을 담당하고 TGT(티켓 발급용 티켓)를 제공한다. 사용자는 이 TGT로 **TGS에 서비스 티켓을 요청**한다. 따라서 인증을 받은 사용자에게 필요한 서비스의 티켓을 발급하는 역할은 TGS에 해당한다.

㉢ 타임스탬프: 시간 정보를 확인해 티켓의 유효기간과 인증 요청의 최신성을 판단한다. 지문의 시간제한에 대응하는 것은 쿠키가 아니라 타임스탬프이다. 다만 **시간제한만으로 모든 재전송이 차단되지는 않는다**. 실제 커버로스에서는 인증자의 타임스탬프와 이미 사용된 인증자를 보관하는 재전송 캐시 등을 함께 검사한다.

따라서 ㉠ 비밀키, ㉡ TGS, ㉢ 타임스탬프로 짝지은 ②가 정답이다.

NoteBase

**문 17. 시스템 침해 공격 유형 및 소프트웨어에 관한 설명으로 가장 적절하지 않은 것은?**

① 크리덴셜 스터핑(credential stuffing)은 사전에 다른 곳에서 습득한 아이디나 패스워드로 로그인을 시도하는 공격이다.

② 스니핑 공격(sniffing attack)은 네트워크상에 전송되는 내용을 도청하여 중요한 정보를 가로채는 공격이다.

③ 랜드 공격(land attack)은 인위적으로 송신자 IP 주소 및 포트 주소를 수신자의 IP 주소 및 포트 주소와 같게 설정하여 시스템 부하를 발생시키는 공격이다.

④ 스파이웨어(spyware)는 컴퓨터 내부의 문서나 사진파일, 스프레드시트 등 사용자의 파일들을 암호화하여 파일을 열지 못하도록 한 뒤, 돈을 요구하는 악성 프로그램이다.

**정답 4번**

**간단한 해설**

파일을 암호화하여 접근을 막고 복구 대가로 돈을 요구하는 것은 **랜섬웨어**이다. 스파이웨어의 주된 특징은 사용자가 모르게 정보를 수집하는 데 있다.

**선지별 해설**

① O: 크리덴셜 스터핑은 다른 서비스 등에서 유출·탈취한 **아이디와 비밀번호의 조합을 재사용**하여 로그인을 시도하는 공격이다. 사용자가 여러 서비스에서 같은 인증정보를 사용하는 점을 악용하며, 임의의 비밀번호를 처음부터 추측하는 방식과 구별된다.

② O: 스니핑은 네트워크를 지나가는 패킷을 몰래 수집·분석하여 통신 내용을 엿보는 공격이다. 암호화되지 않은 인증정보나 메시지 내용은 특히 노출되기 쉽다. 패킷을 수집했다는 사실만으로 적절히 암호화된 내용까지 곧바로 읽을 수 있는 것은 아니다.

③ O: 랜드 공격은 패킷의 출발지와 목적지 IP 주소 및 포트 번호를 같게 위조하여, 취약한 시스템이 자기 자신과 통신하는 것처럼 비정상적인 처리를 하게 만드는 서비스 거부 공격이다. 이 과정에서 자원이 소모되거나 서비스 장애가 발생할 수 있으며, 효과는 대상 시스템의 취약 여부와 방어 기능에 따라 달라진다.

④ X: 사용자의 파일을 암호화해 사용할 수 없게 하고 복구 대가로 돈을 요구하는 악성 프로그램은 **랜섬웨어**이다. 스파이웨어는 사용자가 모르는 사이에 활동이나 정보를 수집하는 악성 프로그램이다. 선지는 정보 수집을 목적으로 하는 스파이웨어와 파일 사용을 막아 금전을 요구하는 랜섬웨어를 혼동했다.

NoteBase

**문 18. 네트워크상에서 안전한 개인금융정보 전송을 위한 SET (Secure Electronic Transaction) 프로토콜에 관한 설명으로 가장 적절하지 않은 것은?**

① 거래인가와 지불기능을 위한 인터페이스 역할을 수행하고 상인의 지불 메시지를 처리하는 장치를 지불 게이트웨이(payment gateway)라고 한다.

② 이중서명은 사용자의 지불정보는 상점에 숨기고 주문정보는 은행에게 숨기는 기능을 제공한다.

③ 암호 프로토콜이 단순하고 RSA 동작으로 프로토콜의 속도가 매우 빠른 장점을 가진다.

④ 카드 소지자에게 전자지갑 소프트웨어를 요구하는 단점을 가진다.

**정답 3번**

**간단한 해설**

SET는 인증서 검증과 전자서명·암호화 등 여러 절차가 결합되어 **구현과 처리가 복잡하다**. RSA 사용은 연산 부담을 수반하므로 프로토콜을 단순하고 매우 빠르게 만드는 장점으로 볼 수 없다.

NoteBase

**선지별 해설**

SET는 인터넷과 같은 공개망에서 카드 결제 정보를 보호하기 위한 프로토콜이다. 카드 소지자·상점·금융기관 사이의 거래에서 암호화, 전자서명, 인증서 검증을 결합한다.

① O: 지불 게이트웨이는 **상점과 금융기관의 결제망을 연결**하는 역할을 한다. 상점이 보낸 지불 메시지와 카드 소지자의 지불 지시를 처리하고, 거래 승인과 대금 지급을 위한 처리를 기존 결제 체계에 연결한다.

② O: SET는 상점에는 주문정보를, 금융기관 측에는 지불정보를 전달하면서 두 정보가 같은 거래에 속함을 확인할 수 있게 한다. 이중서명은 **주문정보와 지불정보의 해시를 하나의 서명으로 연결**하여 각각의 수신자가 상대방 정보 원문을 보지 않고도 연결 관계를 검증하게 한다. 지불정보를 상점이 읽지 못하게 하는 기밀성은 해당 정보의 암호화·분리 전달과 함께 확보된다. 이중서명 자체가 두 원문을 암호화하는 것은 아니다.

③ X: SET는 여러 참여자의 인증서 검증, 전자서명 생성·검증, 암호화와 결제 승인 메시지 교환이 필요해 **절차가 복잡하고 연산 부담이 크다**. RSA는 공개키 기반의 서명이나 키 보호 등에 쓰이며, 이를 사용한다고 프로토콜이 매우 빨라지는 것은 아니다. 메시지 본문에는 대칭키 암호도 함께 사용하는 혼합 방식을 취하므로, SET를 단순하고 RSA 때문에 빠른 프로토콜이라고 설명한 두 부분 모두 부정확하다.

④ O: 카드 소지자는 SET 메시지를 만들고 암호화·서명·검증을 처리하는 **전자지갑 등 전용 소프트웨어**가 필요하다. 이용자가 해당 소프트웨어를 준비하고 거래 정보와 키 등을 관리해야 하므로 도입과 사용의 번거로움이 생긴다.

따라서 복잡한 처리 절차와 공개키 연산 부담을 반대로 설명한 ③이 가장 적절하지 않다.

NoteBase

**문 19. 다음에서 설명하는 접근통제 보안 모델로 가장 적절한 것은?**

자원마다 보안등급을 부여하는 수학적 접근통제 모델로서 군사용 보안구조의 요구사항을 충족하기 위해 설계되었다. 이 모델은 정보의 기밀성에 따라 보안등급을 구분하여 등급이 높은 정보가 등급이 낮은 사용자에게 유출되지 않도록 통제한다. 기밀성 유지에 매우 강한 특성을 가지지만 가용성과 무결성은 고려하지 않는다.

① Role-Based Model

② Task-Based Model

③ Bell-LaPadula Model

④ Lattice-Based Model

**정답 3번**

**간단한 해설**

군사용 보안등급에 따라 정보의 하향 유출을 막는 **기밀성 중심 모델**은 Bell-LaPadula 모델이다. 격자 기반 접근은 이를 포괄하는 일반적 틀이므로, 구체적 모델을 묻는 지문에는 ③이 가장 적절하다.

**핵심 풀이**

핵심 단서: 군사용 다단계 보안 환경에서 보안등급을 이용해 **높은 등급 정보의 하위 등급 유출을 차단**하고 기밀성을 보호한다는 설명은 Bell-LaPadula 모델에 해당한다. 무결성과 가용성까지 함께 보장하는 모델이라는 뜻은 아니다.

통제 원리: 기본 규칙은 주체가 자신보다 높은 보안등급의 객체를 읽지 못하게 하는 **No Read Up**과, 더 낮은 보안등급의 객체에 쓰지 못하게 하는 **No Write Down**이다. 상위 정보를 직접 읽는 행위와, 상위 정보를 하위 객체에 기록해 흘려보내는 행위를 각각 제한한다.

Role-Based Model: 조직의 **역할에 권한을 연결**하고 사용자에게 역할을 부여하는 방식이다. 담당 직무와 역할이 중심이므로 지문의 군사 보안등급에 따른 기밀성 모델을 직접 지칭하는 명칭은 아니다.

Task-Based Model: 수행할 **작업과 업무 진행 상태**에 맞춰 필요한 권한을 관리하는 방식이다. 업무 흐름에 따른 권한의 활성화·변경을 다루므로 지문의 핵심인 기밀 등급 사이의 정보 유출 통제와 구별된다.

Lattice-Based Model: 보안 레이블 사이의 지배 관계를 격자 구조로 표현하여 허용되는 정보 흐름을 정하는 더 일반적인 접근이다. Bell-LaPadula도 이러한 격자 기반 틀로 표현할 수 있으므로 서로 무관한 모델은 아니다. 그러나 지문은 군사 보안과 기밀성 중심이라는 구체적 특징을 함께 제시하므로, 가장 적절한 답은 ③ **Bell-LaPadula Model**이다.

NoteBase

**문 20. 대칭키 암호시스템을 분석하여 대칭키가 190개로 확인되었다. 현재보다 사용 인원을 2배로 확장하면 추가로 필요한 키의 수는 몇 개인가?**

① 1,560

② 590

③ 380

④ 190

**정답 2번**

**간단한 해설**

대칭키 190개는 사용자 20명이 쌍별로 키를 공유하는 경우에 해당한다. 40명으로 늘면 전체 780개가 필요하므로, **기존 190개를 뺀 590개**를 추가해야 한다.

**계산 과정**

키 수 계산의 기준: 사용자 쌍마다 서로 다른 대칭키 하나를 공유하는 방식으로 계산한다. 사용자가 n명이면 각 사람이 나머지 n − 1명과 키를 공유하지만, 이렇게 세면 같은 두 사람의 키를 두 번 세므로 전체 키 수는 $K = \frac{n(n-1)}{2}$개이다. **통신 방향이 달라도 같은 쌍의 키는 하나**로 센다.

현재 사용자 수: 주어진 키가 190개이므로 $\frac{n(n-1)}{2} = 190$, 즉 $n(n-1) = 380$이다. $20 \times 19 = 380$이므로 현재 사용자는 20명이다.

확장 후 전체 키 수: 사용자가 2배가 되면 $20 \times 2 = 40$명이다. 따라서 필요한 전체 키는 $\frac{40 \times 39}{2} = 780$개이다. 키 수는 사용자 수가 아니라 사용자 쌍의 수에 따라 늘어나므로 단순히 190개를 두 배로 계산하면 안 된다.

추가 키 수: 묻는 것은 확장 후 전체가 아니라 **기존 키를 제외하고 더 필요한 수**이다. 따라서 780 − 190 = 590개가 추가로 필요하며, ②에 해당한다.

NoteBase