

# 제133회 정보관리기술사 해설집

2024.05.18

## 국가기술자격 기술사 시험문제

기술사 제 133 회

제 1 교시 (시험시간: 100 분)

분야	정보통신	자격 종목	정보관리기술사	수검 번호		성 명	
----	------	----------	---------	----------	--	--------	--

※ 다음 문제 중 10 문제를 선택하여 설명하시오. (각 10 점)

1. REST API(REpresentational State Transfer Application Programming Interface)에 대하여 설명하시오.
2. 소프트웨어 테스트 유형 중 뮤테이션 테스트(Mutation Test)에 대하여 설명하시오.
3. NoSQL 유형과 모델링 절차를 설명하시오.
4. 전자봉투 생성절차와 개봉절차를 설명하시오.
5. 동형암호(Homomorphic Encryption)의 동작원리와 유형을 설명하시오.
6. 기술수용모델(Technology Acceptance Model: TAM)의 개념과 주요 구성요소에 대하여 설명하시오.
7. 데이터모델링에서 CRUD 매트릭스(Matrix)를 사용하는 목적과 이를 표현하는 방법에 대하여 설명하시오.
8. 인공지능 신뢰성의 개념과 핵심 속성에 대하여 설명하시오.
9. BCP(Business Continuity Planning) 수립 시의 주요 지표와 DRS(Disaster Recovery System) 구축 시의 핵심 고려사항에 대하여 설명하시오.
10. 딥페이크(Deepfake)에 대하여 설명하시오.

11. 소프트웨어 유지보수 향상 및 비용절감을 위한 3R 을 설명하시오.
12. 쿠버네티스(Kubernetes)
13. TCP 프로토콜의 3-way handshake 와 4-way handshake 를 설명하시오.

01	REST API(RePresentational State Transfer Application Programming Interface)		
문제	REST API(RePresentational State Transfer Application Programming Interface)에 대하여 설명하시오.		
도메인	서비스	난이도	중(상/중/하)
키워드	HTTP, URI, POST, GET, PUT, DELETE		
출제배경	클라우드 서비스, MAS, 모바일 애플리케이션 등에서 많이 활용		
참고문헌	ITPE 기술사회 자료		
해설자	모멘텀 안수현 기술사(제119회 정보관리기술사/tino1999@naver.com)		

I. 인터넷상의 정보 교환을 위한, REST API의 개요

가. REST API의 정의

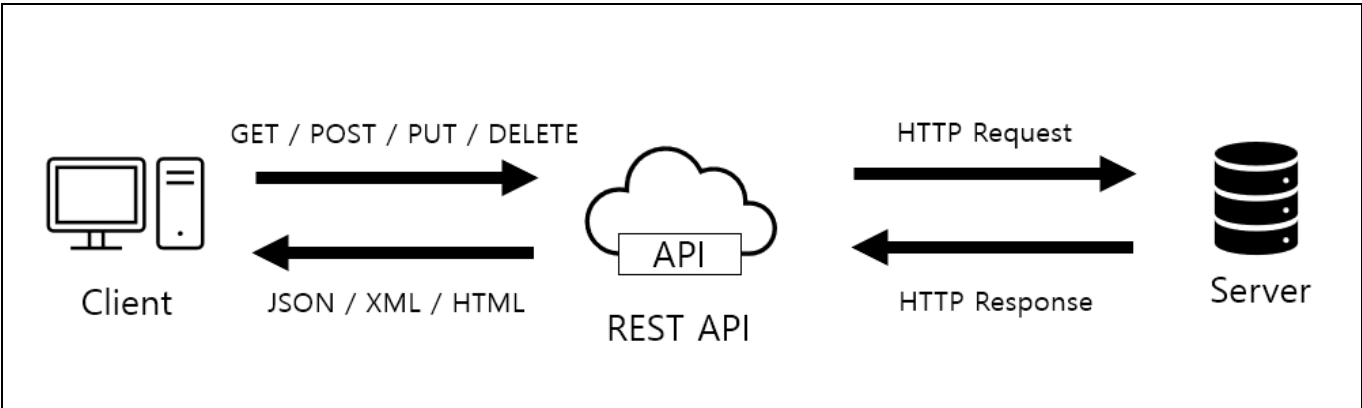
- 인터넷 상에 존재하는 API 대상 자원(데이터 또는 서비스)에 주소를 지정하고, HTTP 통신 프로토콜을 사용하여 해당 자원을 관리하는 아키텍처

나. REST API의 특징

특징	설명
클라이언트-서버 구조	클라이언트와 서버가 명확하게 분리되어 있어 서로의 독립성을 보장
무상태성 (Stateless)	각 요청은 독립적이며 서버는 클라이언트의 상태 정보를 저장하지 않음
캐시 가능 (Cacheable)	HTTP의 특징인 캐시 기능을 적용하여, 응답시간 향상
계층화된 시스템	Client는 REST API Server만 호출하고, REST Server는 다중 계층으로 구성

II. REST API의 개념도 및 구성요소

가. REST API의 개념도



- 자원에 대한 생성·조회·수정·삭제 행위를 HTTP Method(POST, GET, PUT, DELETE)로 표현

나. REST API의 구성요소

구분	항목	세부 내용
구성요소	자원(Resource)	URI를 통해 식별 가능한 데이터 또는 서비스, Open API로 제공하는 대상
	행위(Behavior)	자원에 대한 생성·조회·수정·삭제 행위를 HTTP Method(POST, GET, PUT, DELETE)로 표현

	메시지 (Message)	자원에 대한 행위를 위한 요청정보와 행위에 대한 결과인 응답정보를 출력하는 방식
Method	GET(조회)	기존 자원에 대한 정보 조회 (예시) 사용자 정보 조회: GET https://www.apisample.com/users
	POST(생성)	신규 자원의 생성 (예시) 'itpe'의 사용자 정보 생성: POST https://www.apisample.com/users/itpe
	PUT(수정)	기존 자원에 대한 정보 수정 (예시) 'itpe'의 사용자 정보 변경: PUT https://www.apisample.com/users/itpe
	DELETE(삭제)	기존 자원의 삭제 (예시) 'itpe'의 사용자 정보 삭제: DELETE https://www.apisample.com/users/itpe

- 클라이언트가 서버로부터 데이터를 요청할 때, 서버는 JSON, XML 등의 형식으로 데이터를 전송

### III. REST와 RESTful의 차이점

REST	RESTful
개념적 또는 이론적인 아키텍처 스타일을 말하며, 웹 서비스를 설계하는 데 사용되는 원칙과 가이드라인을 제공	실제 구현을 나타내며, REST 원칙을 충실히 따르는 서비스나 애플리케이션

- 어떤 웹 서비스가 "RESTful"하다는 것은 그 서비스가 REST 아키텍처 스타일을 구현하고 있음을 의미

“끝”

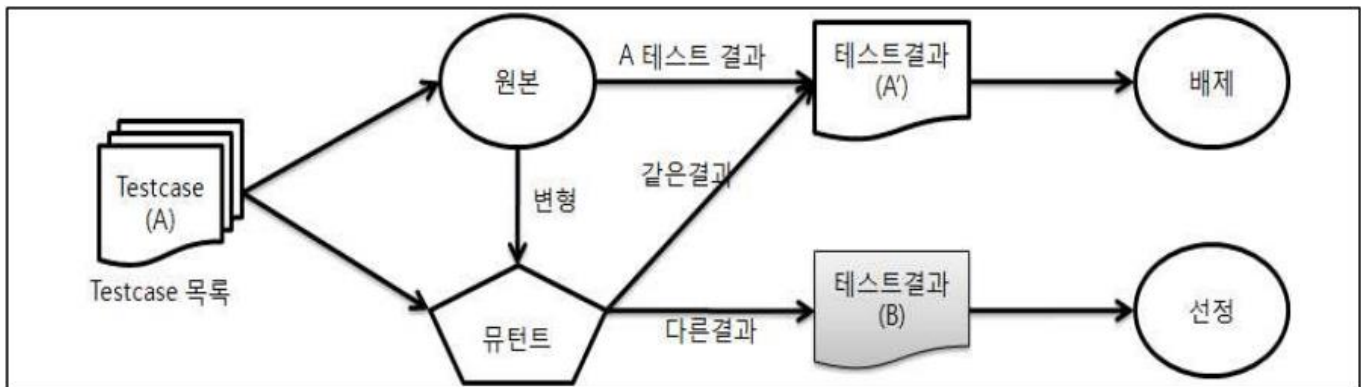
02	뮤테이션 테스트(Mutation Test)		
문제	소프트웨어 테스트 유형 중 뮤테이션 테스트(Mutation Test)에 대하여 설명하시오.		
도메인	SW 공학과 프로젝트관리	난이도	중(상/중/하)
키워드	뮤턴트(Mutant), 뮤턴트 연산자, 뮤테이션 스코어, 비버깅(Bebugging Test)		
출제배경	테스트 케이스의 품질을 극대화 할 수 있는 테스트 기법에 대한 이해도 확인		
참고문헌	ITPE 기술사회 자료		
해설자	모멘텀 안수현 기술사(제119회 정보관리기술사/tino1999@naver.com)		

### I. 의도된 결함으로 테스트 데이터의 효과성 검증, Mutation Test 개요

- 결함 추정과 테스트 케이스를 평가하기 위해 프로그램의 구문을 일정한 규칙을 적용하여 변형(Mutant)후 원래 프로그램과 동일한 케이스로 테스트하는 결함기반 테스트 기법

### II. Mutation Test 수행 절차 및 프로세스

#### 가. Mutation Test 수행 절차



- 원본 소스코드에 속성, 서비스, 입력값, 출력값을 변경시켜 테스트를 수행하며, 결과값에 대한 비교를 통해 테스트 데이터 신뢰성 검증

#### 나. Mutation Test 프로세스

절차	설명
컴포넌트 변용	- 컴포넌트 선택, 변용, 통합 단계 중 Mutation Test를 위한 변용 작업 수행
오류 형태 파악	- 검출하고자 하는 오류 형태를 파악(사용자 변용 오류, 컴포넌트 자체 결함)
연산자 결정	- 속성 변용, 서비스 변용, 입력 값 변용, 출력값 변용에 필요한 연산자 결정
테스트 수행	- 식별된 결함을 피드백 하여 컴포넌트의 재 변용 및 재시험 실시
결과 분석	- 테스트 케이스가 뮤턴트의 변경을 감지하면, 테스트 케이스는 유효한 것으로 간주- - 뮤턴트가 테스트를 통과하면, 해당 테스트 케이스는 추가적인 개선이 필요

- 소프트웨어 코드에 의도적으로 결함이나 버그를 삽입후 이를 제거하는 비버깅(Bebugging)도 활용

### III. 뮤테이션 테스트와 디버깅 테스트 비교

항목	뮤테이션 테스트	디버깅 테스트
목적	테스트 케이스의 효율성과 커버리지 평가	잔존 오류 측정 및 디버깅 기술의 향상과 검증
수행 방식	돌연변이 생성	의도적 오류코드 삽입
적용 대상	테스트 프로세스 자체를 검증하고 강화하는 데 사용	디버깅 과정과 도구를 개선하는 데 활용
결과 활용	테스트 케이스의 미비한 부분을 발견하고 이를 강화시키는 데 사용	프로그래머가 직접적으로 버그를 찾아내고 해결하는 기술을 연습하고 평가

“끝”

03	NoSQL		
문제	NoSQL 유형과 모델링 절차를 설명하시오.		
도메인	DB와 데이터관리	난이도	중(상/중/하)
키워드	CAP, PACELC, Key-Value, Document, Column, Graph		
출제배경	빅 데이터, 실시간 웹 애플리케이션, IoT 등 다양한 분야에서 NoSQL의 활용이 증가		
참고문헌	ITPE 기술사회 자료		
해설자	모멘텀 안수현 기술사(제119회 정보관리기술사/tino1999@naver.com)		

I. RDBMS의 복잡도와 용량 한계 극복, NoSQL의 개념

CAP 이론	정의
<p>분산 환경에서 모든 Server가 같은 시점에 동일한 데이터를 바라보는 것을 의미</p> <p>클러스터를 구성하는 일부 Server가 다운 되도 데이터베이스 시스템은 정상적 동작을 의미</p> <p>네트워크 장애로 인하여 Message가 전달되지 않거나 유실되었을 때 시스템이 동작하는 메커니즘</p> <p>Oracle MySQL DB2 → RDBMS</p> <p>HBase MongoDB → NoSQL</p> <p>Dynamo Cassandra</p>	<p>- 관계형 데이터베이스(RDBMS)의 테이블-컬럼과 같은 스키마 없이, 분산 환경에서 단순 검색 및 추가 작업이 용이하고, 지연(latency)과 처리율(throughput)이 높은 Database</p>

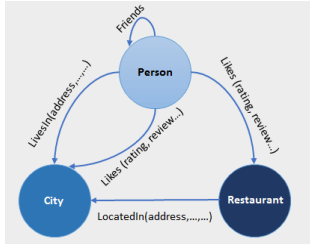
- CAP이론에 기반하고 있지만, 최근 CAP 이론을 개선한 PACELC 이론도 등장

II. NoSQL의 유형과 모델링 절차

가. NoSQL의 유형

모델 유형	개념도	설명
Key-Value		<p>- 키와 값의 쌍으로 관리</p> <p>- 키를 사용 값을 확인하는 가장 기본적인 단순한 NoSQL 데이터 모델</p> <p>- 단순한 모델 이므로 Key에 대한 단위연산 속도가 빠름</p> <p>- 키 범위(Key Range)처리 요구에 대한 적용이 안 되는 단점</p>
Ordered Key-Value (Column Family)		<p>- 키 범위 처리 요구를 개선한 Key-Value 데이터 모델</p> <p>- 키를 기반으로 Sorting 하여 저장</p>



Document Key/Value Store	<table border="1"> <tr> <td>Key</td> <td>JSON/XML Document</td> </tr> <tr> <td>Key</td> <td>JSON/XML Document</td> </tr> </table>	Key	JSON/XML Document	Key	JSON/XML Document	<ul style="list-style-type: none"> <li>- 스키마 없이 임의의 속성을 추가할 수 있는 데이터 모델</li> <li>- Document Id 또는 특정 속성값 기준으로 order-preserving 하여 키 범위 처리에 대한 효율적인 연산이 가능</li> <li>- JSON, XML 형태로 구조적 문서 저장 가능</li> <li>- 쿼리 처리시 데이터를 Parsing 하여 메모리 연산을 해야 하므로 처리 overhead가 Key-Value 데이터 모델에 비해 큼</li> </ul>
Key	JSON/XML Document					
Key	JSON/XML Document					
Graph 기반		<ul style="list-style-type: none"> <li>- Entities와 Entities 간의 관계(relations)를 저장하는 데이터 베이스 유형</li> <li>- 테이블이 느슨하게 연결된 RDBMS와 비교할 때, Graph 기반 DB는 본질적으로 다중 관계형 DB임.</li> <li>- 모든 Node(Vertex)와 Edge에 고유 식별자가 부여됨.</li> <li>- Traversing relationship가 이미 DB에 저장되어 있어서 추가적인 계산이 필요 없기 때문에 검색 속도가 빠름.</li> </ul>				

#### 나. NoSQL 모델링 절차

단계	절차	설명
탐색	도메인 모델 파악	<ul style="list-style-type: none"> <li>- 저장하고자 하는 도메인 파악</li> <li>- 어떤 데이터 개체가 있는지 개체 간의 관계는 어떻게 되는지 등을 분석하고 ERD활용하여 도식화</li> </ul>
설계	쿼리결과 디자인	- 도메인 모델 기반으로 애플리케이션에 의해서 쿼리 되는 결과값을 정하여 데이터 출력 내용 기반으로 디자인.
	패턴을 이용한 데이터 모델링	- RDBMS 기능을 제공하지 않기 때문에 이를 배제하고 Put/Get으로만 데이터를 가지고 올 수 있는 형태로 데이터 모델, NoSQL 내의 테이블로 재정의.
	기능 최적화	- RDBMS의 Index 와 같은 이 개념을 NoSQL 에서 'Secondary Index'을 이용하여 기능의 최적화
최적화	후보 NoSQL을 선정 및 테스트	- NoSQL에 대한 구조 및 특성을 분석한 후에 실제로 부하테스트, 안정성, 확장성 테스트를 거친 후에 가장 적절한 솔루션을 선택.
	선정된 NoSQL의 데이터 모델 최적화 및 하드웨어 디자인	- 선정된 NoSQL 을 기반으로 그에 적합한 데이터 모델 최적화 후 이에 맞는 어플리케이션 인터페이스 설계와 구동시킬 하드웨어 디자인을 실시

- NoSQL의 key-value 구조로서 RDBMS와 차이점이 존재하며, 모델링 절차 시 구조와 특성의 고려가 필요

III. NoSQL 모델링 사례

구분	모델링 사례		
RDBMS 모델링	<div> <div>평론가</div> <div> <div>평론가 ID</div> <div>이름</div> <div>Blog 주소</div> </div> </div> <div> <div>영화평</div> <div> <div>평론가 ID (FK)</div> <div>영화ID(FK)</div> <div>한줄영화평</div> </div> </div> <div> <div>영화</div> <div> <div>영화 ID</div> <div>영화제목</div> </div> </div>		
NoSQL 모델링	<div> <div>Table</div> <div>영화 평론가</div> </div> <div> <div>Row key</div> <div>평론가 ID</div> </div> <div> <div>Columns</div> <div> <div>이름 string</div> <div>Blog 주소 string</div> </div> </div> <div> <div>영화평</div> <div> <div>영화ID byte[10]</div> <div>영화제목 string</div> <div>한줄영화평 string</div> </div> </div>		

“끝”

04	전자봉투		
문제	전자봉투 생성절차와 개봉절차를 설명하시오.		
도메인	보안	난이도	중(상/중/하)
키워드	전자서명, 공개키, 비밀키, Hash, Message Digest		
출제배경	개인정보 보호 및 데이터 보안을 보장하는 기술 확인		
참고문헌	ITPE 기술사회 자료		
해설자	모멘텀 안수현 기술사(제119회 정보관리기술사/tino1999@naver.com)		

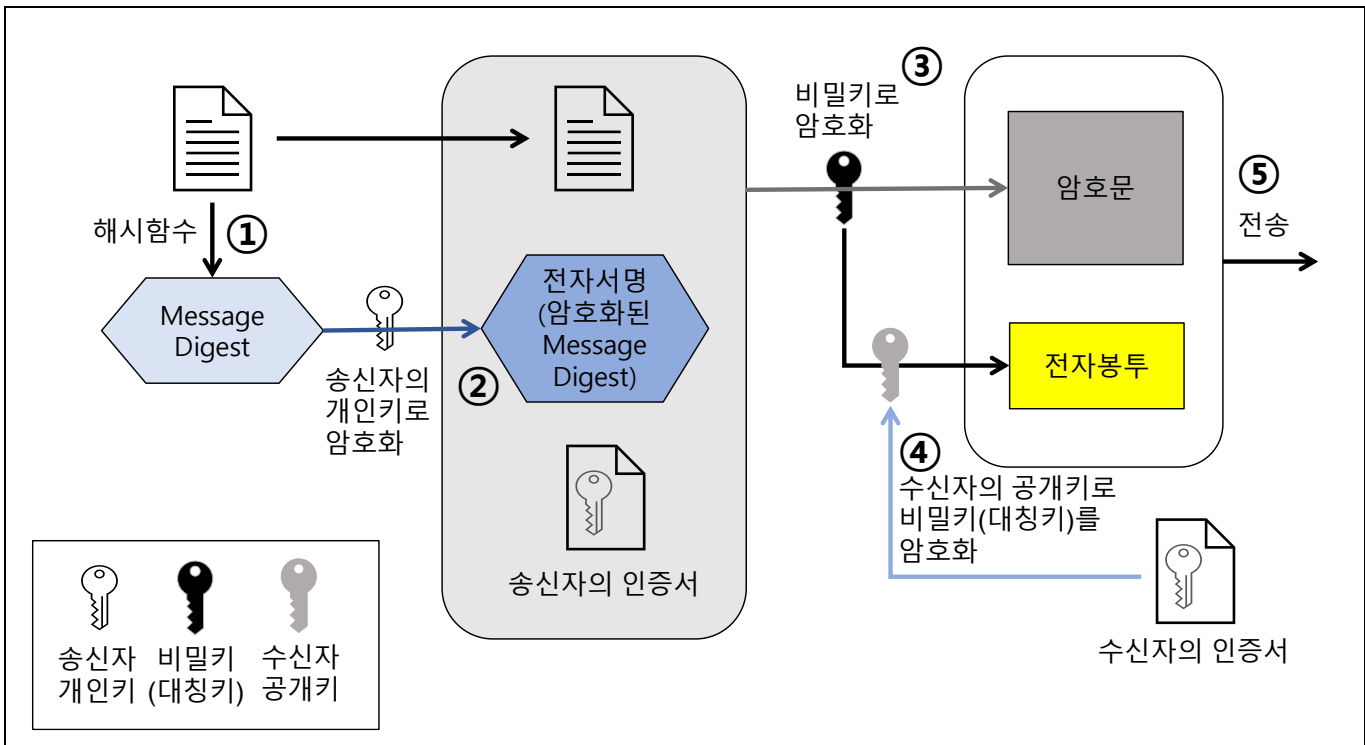
I. 전자상거래에서의 보안기술, 전자봉투의 개념

개념	- 공개키 인프라를 이용해 비밀키를 수신자의 공개키로 암호화함으로써 온라인 상에서 기밀성 있는 통신이 가능한 암호화 기술	
특징	- 기밀성	- 허락되지 않은 사용자의 정보 내용을 알 수 없도록 하는 것(비밀 보장) - 원치 않는 정보의 공개를 막는다는 의미에서 프라이버시 보호와 밀접함
	- 전자서명 특징 포함	- 무결성, 부인방지 모두 지원 - 공개키 암호화 알고리즘 및 해시 함수 사용
	- 대칭키 암호화	- 송신자 대칭키(Secret Key)를 수신자의 공개키로 암호화

- 대칭키(비밀키)의 빠른 암호화 속도와 비대칭키(공개키)의 키 관리 장점을 이용한 메시지 전달 기법

II. 전자봉투를 활용한 전자서명의 송/수신 절차 상세 설명

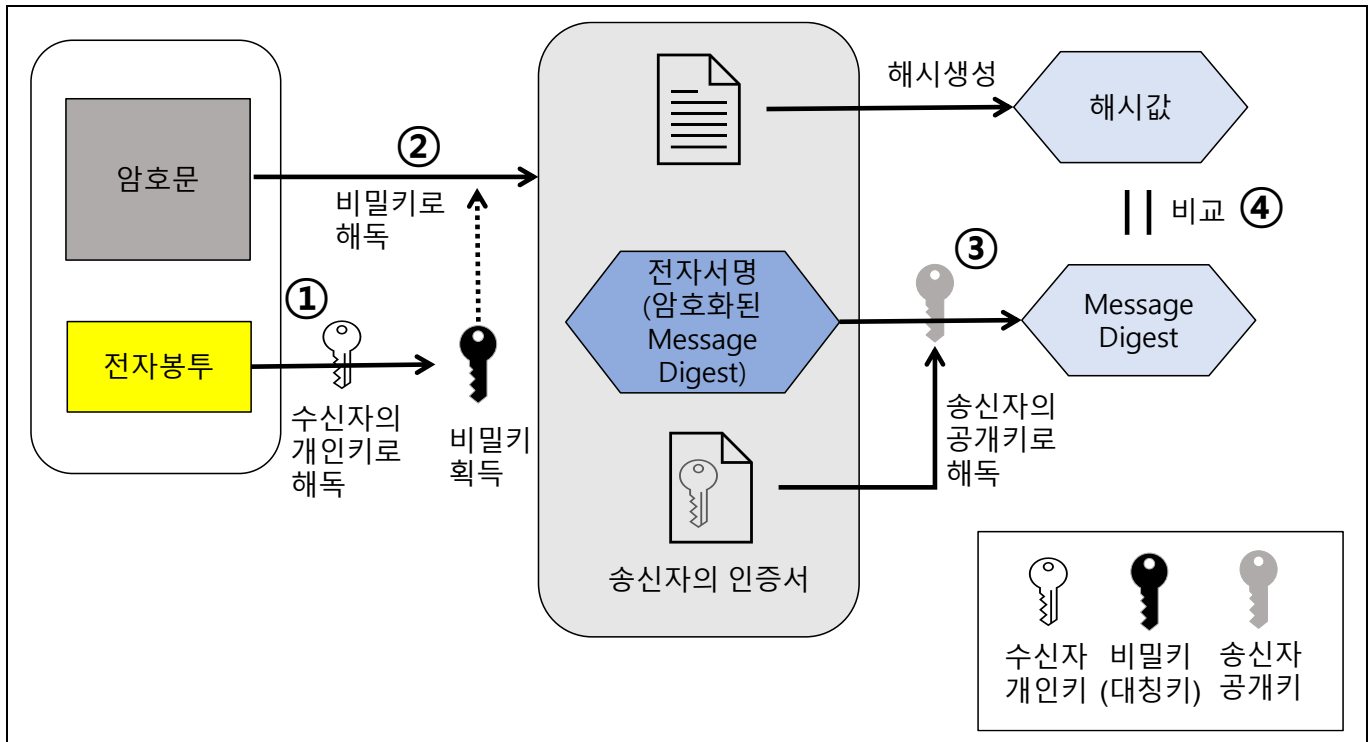
가. 전자봉투를 활용한 전자서명 송신 절차 상세 설명



절차	설명
1. Message Digest 생성	- 메시지를 해시함수로 축약, Message Digest 생성
2. 전자서명 생성	- 송신자의 개인키(Private Key)로 Message Digest를 암호화 한 전자서명 생성
3. 비밀키(Secret Key)로 암호문 생성	- 비밀키(대칭키)를 사용하여 전자서명과 메시지와 송신자의 공개키가 들어 있는 인증서를 암호화하여 암호문 생성
4. 전자봉투 생성	- 수신자의 공개키(Public Key)로 비밀키를 암호화하는데, 이 암호화된 비밀키가 전자봉투가 됨
5. 암호문과 전자봉투 전송	- 송신자는 최종적으로 비밀키로 암호화된 전자서명과 전자봉투를 수신자에게 전송

- 송신자의 개인키로 전자서명을 생성, 수신자의 공개키로 전자봉투를 생성하여 전송

#### 나. 전자봉투를 활용한 전자서명 수신 절차 상세 설명



절차	설명
1. 전자봉투 복호화	- 전달받은 전자봉투를 수신자의 개인키로 복호화 하여 비밀키(대칭키)를 획득
2. 암호문 복호화	- 비밀키(대칭키)를 이용해 암호문을 복호화 하여 전자서명과 메시지, 송신자의 인증서를 획득
3. 전자서명 복호화	- 획득한 송신자 인증서의 송신자 공개키를 이용해 전자서명을 복호화 하여 Message Digest 획득
4. 무결성 검증	- 메시지의 해시값과 Message Digest를 비교해 무결성 검증

- 전자봉투를 통해 전자서명의 위조불가, 부인방지, 서명자 인증, 변경불가, 재사용 불가와 함께 기밀성 제공

## III. 전자서명과 전자봉투에 사용되는 기술요소

기술 요소	알고리즘	설명
공개키 (비대칭키)	- RSA, KCDSA	- 암호화키 ≠ 복호화키 - 개인키/공개키 기반 문서검증, 대칭키에 비해 암호화 속도 느림
비밀키 (대칭키)	- DES, SEED, AES, ARIA, RC4	- 암호화키 = 복호화키 - 암호화 속도 빠름, 경제성 높음, 키 분배 및 관리 불편
해시 함수	- SHA-1, SHA-256, SHA-512, MD5, HAS	- 임의 길이를 갖는 메시지를 입력 받아 고정된 길이의 해시값을 출력하는 함수 - Message Digest 생성, 메시지 무결성 검증

- 전자봉투를 활용한 전자서명 기술은 SSL보안 인증서, 공인인증서, 전자상거래를 위한 이중서명 등에 사용됨.

“끝”

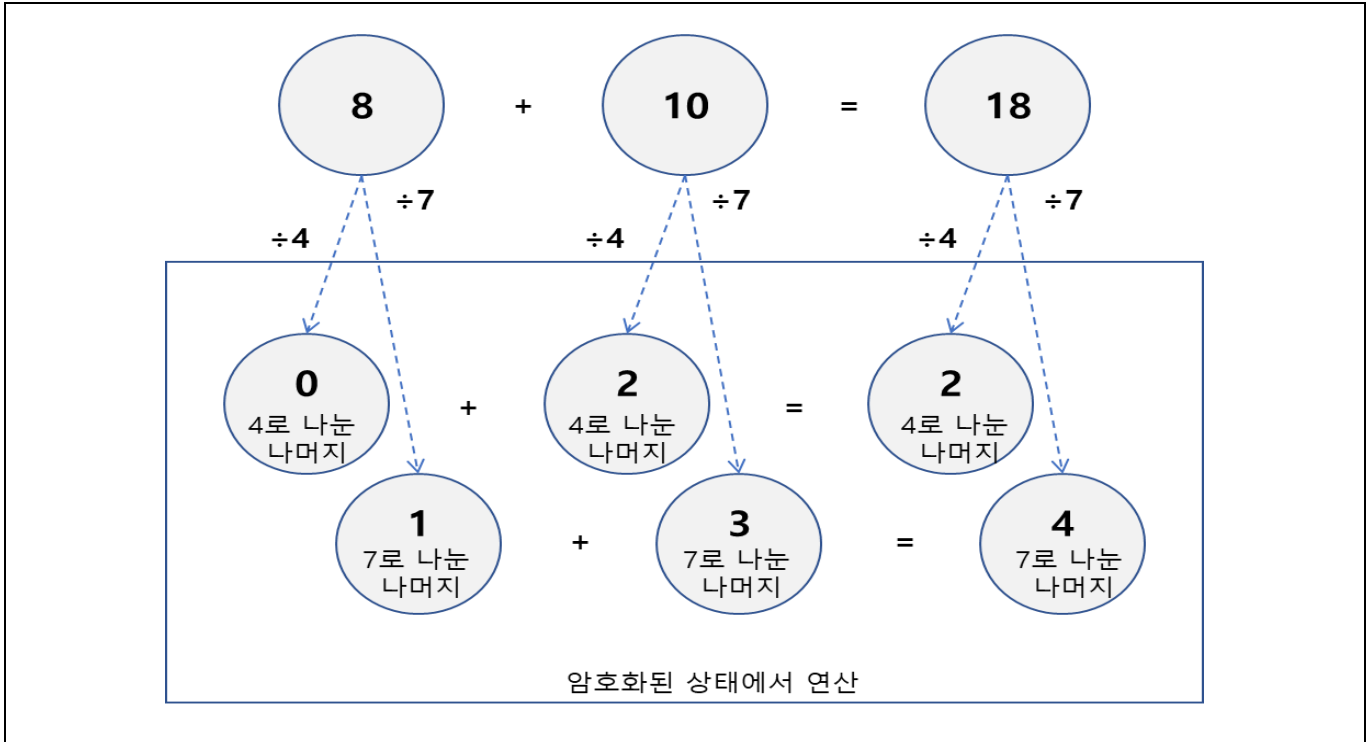
05	동형암호		
문제	동형암호(Homomorphic Encryption)의 동작원리와 유형을 설명하시오.		
도메인	정보보안	난이도	중 (상/중/하)
키워드	스쿼싱, 부트스트래핑, 부분동형암호, 준동형암호화, 완전동형암호화		
출제배경	4세대 암호화 기술 이해 확인		
참고문헌	ITPE 기술사회 자료		
해설자	NS반 김민재 기술사(제 124회 정보관리기술사 / kmj_pe@naver.com)		

I. 차세대 암호기술, 동형암호의 개념

- 평문에 대한 연산을 수행한 후 암호화한 결과(암호문)와 각각의 암호문에 대하여 연산을 수행한 결과가 같은 값을 가지는 암호화 방식

II. 동형암호의 동작원리 및 기술요소

가. 동형암호의 동작원리 개념도



- 평문의 연산 결과 값과 암호화된 상태에서의 연산 결과 값이 동일

### 나. 동형암호의 동작원리 설명

구분	기술요소	설명
동작원리	- 부분 동형 암호 (SHE)	- 덧셈과 곱셈 모두 지원 - 곱셈은 특정 횟수 K번까지만 지원하도록 설계
	- 스쿼싱 (Squashing)	- 노이즈 증가 감소, 평문 변형되지 않도록 구성하는 원리 - 복호화 알고리즘 및 공개키 일부 변형 - 곱셈 k회 이하만 적용시 복호화 가능 - 복호화 알고리즘이 동형으로 계산 가능
	- 부트스트래핑 (Bootstrapping)	- 암호문에 암호화된 비밀키를 이용하여 노이즈가 감소된 새로운 암호문 생성 후 연산 수행하는 설계원리 - 암호화된 키와 이중 암호화된 암호문 입력한 후 복호화 알고리즘 실행 - 새로운 암호문 생성(Recrypt) - 동형 연산 후 매번 Recrypt 수행하여 FHE를 생성
특성	- 서킷 프라이버시	- 연산 진행 시 연산에 대한 정보를 알지 못하는 성질
	- 다중 도약 동형성	- 생성된 암호문이 다른 동형 연산의 입력으로 사용이 가능한 성질
	- 보안성	- 암호화된 형태로 연산이 진행되어 해킹 차단할 수 있는 성질

- 부분동형암호, 스쿼싱, 부트스트래핑을 이용해서 완전동형암호를 구현

### III. 동형암호의 유형

유형	특징	설명
부분적 동형암호	한 유형 연산	- Partial homomorphic encryption (PHE) - 오직 한유형의 연산만 지원
준 동형암호	덧셈(뺄셈) 또는 곱셈 (나눗셈) 가능	- Somewhat homomorphic encryption (SHE) - 암호화된 상태에서 제한된 회수의 특정 연산 수행 가능 - 덧셈과 곱셈 모두 지원하며 몇번의 연산에도 값 보존 처리 - 연산 회수 제한 - 빠른 속도 연산
완전 동형함수	사칙연산 모두 가능	- Fully homomorphic encryption (FHE) - 암호화된 상태에서 암호문만으로 평문의 연산이 가능 - 준동형암호에 bootstrapping을 통해 이론상으로 대수연산(덧셈, 곱셈, XOR)을 무한번 수행 가능 - 사칙 연산 모두 사용 가능 - 무제한 연산 가능 - 큰 용량으로 인해 속도 저하

“끝”

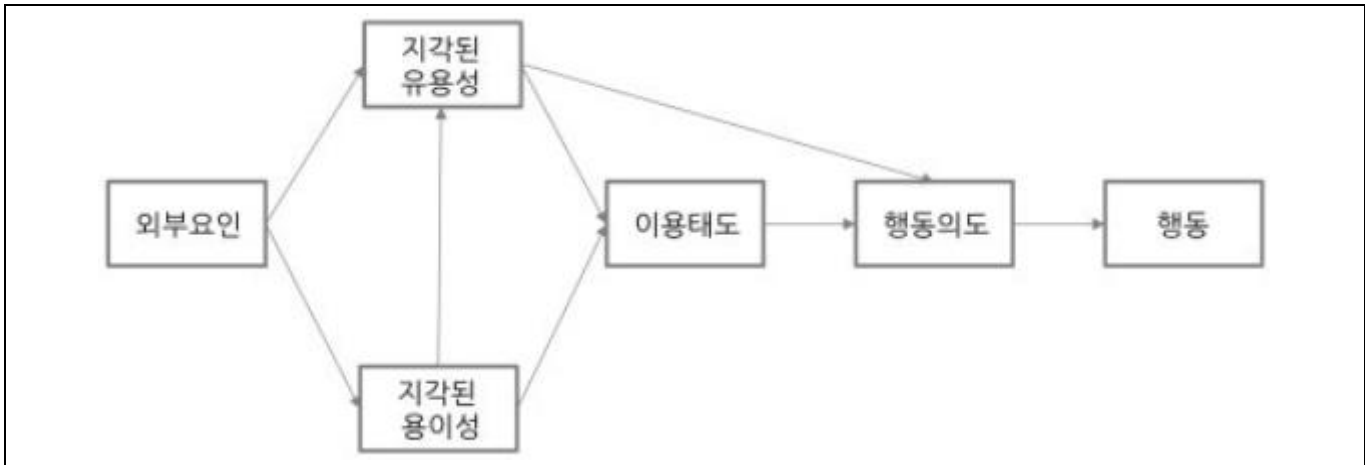
06	기술수용모델		
문제	기술수용모델(Technology Acceptance Model: TAM)의 개념과 주요 구성요소에 대하여 설명하시오.		
도메인	디지털서비스	난이도	상 (상/중/하)
키워드	유용성, 용이성, 이용 태도, 행동		
출제배경	정보기술 시스템에서 이용자의 행위 설명을 위한 모델 이해 확인		
참고문헌	ITPE 기술사회 자료 <a href="https://brunch.co.kr/@linecard/702">https://brunch.co.kr/@linecard/702</a>		
해설자	NS반 김민재 기술사(제 124회 정보관리기술사 / kmj_pe@naver.com)		

I. 기술수용모델 (TAM) 개념

- 사회 심리학 분야의 합리적 행위 이론(TRA, Theory of Reasoned Action)에 기반하여 정보기술 이용자의 행위를 설명하고 예측하는 모델 (사람들이 신기술에 기반한 혁신 제품을 수용하는 과정을 설명)

II. 기술수용모델 구성 요소

가. 기술수용모델 구성도



- 주요 4 가지 변수로 기술 수용 모델 구성

나. 기술수용모델 구성 요소 상세 설명

구성 요소	특징	설명
인지된 유용성 (PU, Perceived Usefulness)	직무 유용성	- 정보기술이 개인의 직무 향상에 도움을 주는 정도 - 정보기술이 개인의 업무 수행 능력과 삶의 질이 향상될 것이라는 결과에 대한 평가
인지된 용이성 (PEOU, Perceived Ease of Use)	활용 용이성	- 정보기술을 사용하는 것이 쉽다고 믿는 정도 - 정보기술을 습득하는 개인의 정신적 또는 물리적 노력이 들지 않는 정도
이용 태도 (A, Attitude Toward Using)	행동 의도	- 개인의 행동에 대한 신념과 감정을 의미 - 직접적으로 행동 의도에 영향을 미침



행동 (BI, Behavioral Intention to Use)	구매 및 평가	- 실제 구매 또는 사용 - 긍정적 또는 부정적 평가
--------------------------------------	---------	----------------------------------

- 신기술에 대한 잠재적 수용자가 지닌 기존 가치관과 과거의 경험 및 필요 여부를 인식되는 **적합성과 외부 요인을 구체화하지 않은 단점** 해결을 위해 기술 수용 모델이 TAM2, TAM3로 발전

### III. 발전된 기술 수용 모델

#### 가. 확장된 기술 수용 모델 (TAM2, Extended Technology Acceptance Model)

개념	- 기존 TAM 모델에 조직 차원의 정보기술 수용 과정에 영향을 미치는 외부 요인들을 포함하고 이용태도를 제외한 확장된 기술 수용 모델	
5 가지 외부요인	주관적 규범	- 조직 구성원들이 자사의 행위에 대해 어떻게 생각할 지에 대한 인식
	사회적 이미지	- 혁신을 채택하여 사회에서 자신의 이미지 또는 지위를 높일 것으로 인식하는 정도
	업무 관련성	- 정보기술을 자신의 업무에 활용할 수 있다고 믿는 정도
	결과 품질	- 정보기술을 자신의 업무를 개선한다고 믿는 정도
	결과 입증 가능성	- 정보기술을 활용한 결과가 분명하고 관찰 가능하고 결과를 전달할 수 있다고 믿는 정도

#### 나. 포괄적 기술 수용 모델 (TAM3, Extended Technology Acceptance Model)

개념	- 조직 구성원들의 의사결정에 관한 통합 모델	
6 가지 외부요인	컴퓨터 자기 효능감	- 컴퓨터를 활용하여 특정 업무나 수행할 수 있는 능력을 가졌다고 믿는 정도
	외부지원 인식	- 특정 시스템 이용을 지원해주는 조직 차원의 기술 지원이 있다고 믿는 정도
	컴퓨터 불안	- 컴퓨터를 이용해야 하는 상황에서 개인이 느끼는 거정이나 공포의 정도
	컴퓨터 유희성	- 컴퓨터를 이용한 상호 작용에 대한 인지적인 자발성 정도
	인지된 즐거움	- 특정 시스템을 이용하는 행위 자체가 즐겁다고 인식하는 정도
	객관적인 용이성	- 특정 업무를 수행하는 데 요구되는 노력의 실제 수준

“끝”

07	데이터 모델링		
문제	데이터모델링에서 CRUD 매트릭스(Matrix)를 사용하는 목적과 이를 표현하는 방법에 대하여 설명하시오.		
도메인	DB와 데이터관리	난이도	상 (상/중/하)
키워드	단위 프로세스, 엔티티 타입, 생성(Create), 이용(Read), 수정(Update), 삭제>Delete)		
출제배경	CRUD 매트릭스 이해도 확인		
참고문헌	ITPE 기술사회 자료		
해설자	NS반 김민재 기술사(제 124회 정보관리기술사 / kmj_pe@naver.com)		

I. 프로세스와 데이터 간의 상관관계 표현을 위한 Matrix, CRUD Matrix의 개요

가. CRUD Matrix의 정의

- 시스템 개발 시 프로세스(또는 메소드, 클래스)와 DB에 저장되는 데이터 사이의 Dependency를 나타내기 위한 Matrix

나. CRUD Matrix의 목적

목적	설명
모델링 작업 검증	- 분석 단계의 데이터 모델과 프로세스 모델에 대한 작업을 검증하는 역할
중요 산출물	- 시스템 구축 단계에서 어플리케이션 개발 시 필요하며 중요한 산출물
테스트에서 사용	- 테스트 단계에서 개발한 어플리케이션 테스트를 위한 객관적인 자료
인터페이스 현황 파악	- 전체 업무의 인터페이스를 파악

II. CRUD Matrix 표현 방법 및 구성 요소

가. CRUD Matrix 표현 방법

단위프로세스 엔티티타입	고객	대여 주문	대여 목록	연 체 료	책 목 록
	책 대여주문을 접수한다.	R	C		
	책을 대여한다.		R	C	R
	책 대여불가를 통보한다.		U		R
	연체료를 등록한다.	R		R	C
	대여목록 기간을 수정한다.			U	
	책을 반납한다.		U	U	

- 행은 업무 프로세스, 열은 엔티티 타입으로 구성되며 생성(C), 이용(R), 수정(U), 삭제(D) 상태를 표시

나. CRUD Matrix 구성 요소

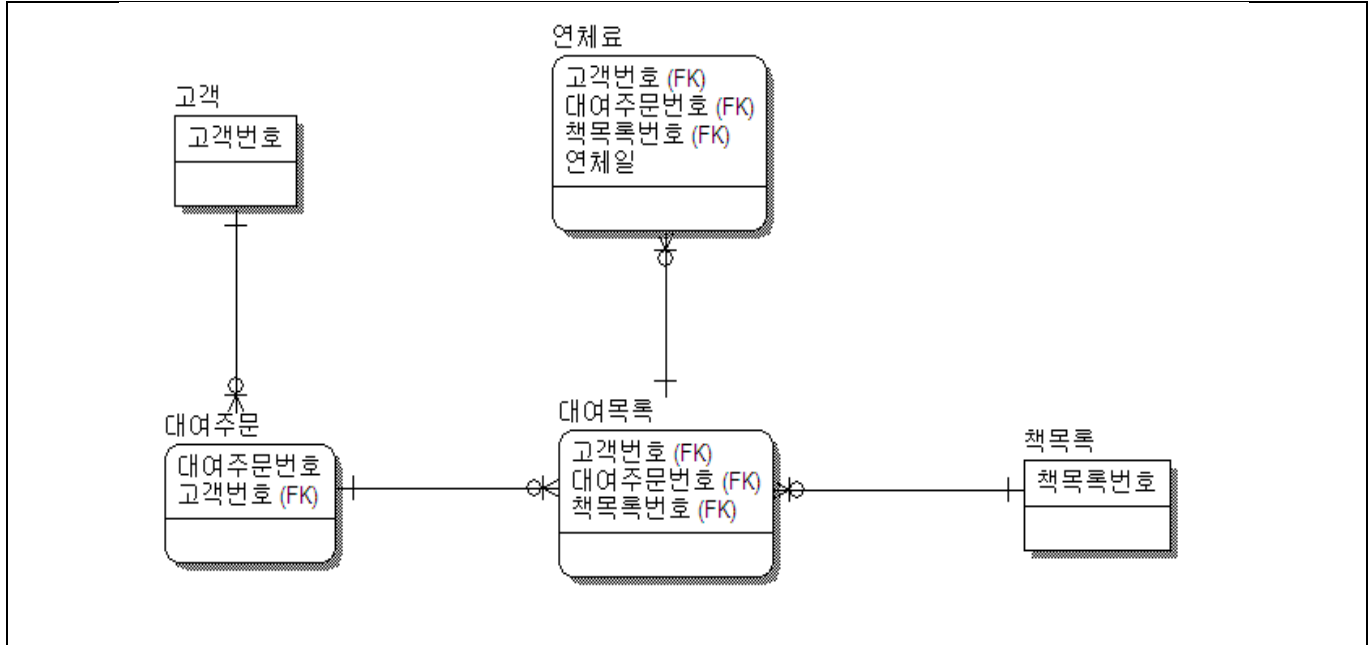
구성 요소	설명	사례
엔티티 타입	- 프로세스에 영향을 받는 데이터	고객, 제품, 주문, 목록

단위 프로세스	- 엔티티에 영향주는 업무 단위	고객 등록, 주문 조회
CRUD	- 프로세스가 엔티티에 주는 영향, Create, Read, Update, Delete	조회, 생성, 수정, 삭제

- 엔티티 타입은 데이터 베이스 튜플 역할, 단위 프로세스는 데이터 베이스 트랜잭션 역할 수행

### III. CRUD Matrix 작성 절차

#### 가. 데이터 모델링 : ERD 분석



#### 나. 업무 시나리오 작성 : 업무기능 분해도 작성

Level 1	Level 2	Level 3
책 대여관리	주문접수	책 대여주문을 접수한다.
		책을 대여한다.
		대여불가를 알려준다.
	대여기간 연기	연체료를 등록한다.
		대여목록기간을 수정한다
	반납	책을 반납한다.

#### 다. CRUD Matrix 작성

단위프로세스 \ 엔티티타입	고객	대여주문	대여목록	연체료	책목록
책 대여주문을 접수한다.	R	C			
책을 대여한다.		R	C		R
책 대여불가를 통보한다.		U			R
연체료를 등록한다.	R		R	C	
대여목록 기간을 수정한다.			U		
책을 반납한다.		U	U		

- CRUD Matrix 작성 시 5가지 주요 점검 사항 확인 필요

IV. CRUD Matrix 작성 시 5가지 주요 점검 사항

점검 사항	내용																																													
1. 모든 엔티티 타입은 CRUD가 한 번 이상 표기되었는가?	<table><tr><th>단위프로세스 \ 엔티티타입</th><th>고객</th><th>주문</th><th>사원</th><th>주문목록</th><th>제품</th></tr><tr><td>신규고객을 등록한다</td><td>C</td><td></td><td></td><td></td><td></td></tr><tr><td>주문을 신청한다</td><td>R</td><td>C</td><td></td><td>C</td><td>R</td></tr><tr><td>주문량을 변경한다</td><td></td><td>R</td><td></td><td>U</td><td></td></tr><tr><td>주문을 취소한다</td><td></td><td>D</td><td></td><td>D</td><td></td></tr><tr><td>제품을 등록한다</td><td></td><td></td><td></td><td>C</td><td></td></tr><tr><td>고객정보를 조회한다</td><td></td><td></td><td></td><td></td><td></td></tr></table> <p>엔티티타입에 발생하는 CRUD가 존재하지 않음</p>	단위프로세스 \ 엔티티타입	고객	주문	사원	주문목록	제품	신규고객을 등록한다	C					주문을 신청한다	R	C		C	R	주문량을 변경한다		R		U		주문을 취소한다		D		D		제품을 등록한다				C		고객정보를 조회한다								
단위프로세스 \ 엔티티타입	고객	주문	사원	주문목록	제품																																									
신규고객을 등록한다	C																																													
주문을 신청한다	R	C		C	R																																									
주문량을 변경한다		R		U																																										
주문을 취소한다		D		D																																										
제품을 등록한다				C																																										
고객정보를 조회한다																																														
2. 모든 엔티티 타입에 "C"가 한 번 이상 존재하는가?	<table><tr><th>단위프로세스 \ 엔티티타입</th><th>고객</th><th>주문</th><th>주문목록</th><th>제품</th></tr><tr><td>신규고객을 등록한다</td><td>C</td><td></td><td></td><td></td></tr><tr><td>주문을 신청한다</td><td>R</td><td>C</td><td>C</td><td>R</td></tr><tr><td>주문량을 변경한다</td><td></td><td>R</td><td>U</td><td></td></tr><tr><td>주문을 취소한다</td><td></td><td>D</td><td>D</td><td></td></tr><tr><td>제품을 등록한다</td><td></td><td></td><td>C</td><td></td></tr><tr><td>고객정보를 조회한다</td><td>R</td><td></td><td></td><td></td></tr></table> <p>엔티티타입에 조회 (READ)는 있는데, 데이터를 생성하는 단위 프로세스가 없음</p>	단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품	신규고객을 등록한다	C				주문을 신청한다	R	C	C	R	주문량을 변경한다		R	U		주문을 취소한다		D	D		제품을 등록한다			C		고객정보를 조회한다	R													
단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품																																										
신규고객을 등록한다	C																																													
주문을 신청한다	R	C	C	R																																										
주문량을 변경한다		R	U																																											
주문을 취소한다		D	D																																											
제품을 등록한다			C																																											
고객정보를 조회한다	R																																													
3. 모든 엔티티 타입에 "R"이 한 번 이상 존재하는가?	<table><tr><th>단위프로세스 \ 엔티티타입</th><th>고객</th><th>주문</th><th>주문목록</th><th>제품</th></tr><tr><td>신규고객을 등록한다</td><td>C</td><td></td><td></td><td></td></tr><tr><td>주문을 신청한다</td><td>R</td><td>C</td><td>C</td><td>R</td></tr><tr><td>주문량을 변경한다</td><td></td><td>R</td><td>U</td><td></td></tr><tr><td>주문을 취소한다</td><td></td><td>D</td><td>D</td><td></td></tr><tr><td>제품을 등록한다</td><td></td><td></td><td></td><td>C</td></tr><tr><td>고객정보를 조회한다</td><td>R</td><td></td><td></td><td></td></tr></table> <p>주문목록은 생성하고, 수정하고, 삭제는 하지만 데이터가 읽혀지지 않는다.</p>	단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품	신규고객을 등록한다	C				주문을 신청한다	R	C	C	R	주문량을 변경한다		R	U		주문을 취소한다		D	D		제품을 등록한다				C	고객정보를 조회한다	R													
단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품																																										
신규고객을 등록한다	C																																													
주문을 신청한다	R	C	C	R																																										
주문량을 변경한다		R	U																																											
주문을 취소한다		D	D																																											
제품을 등록한다				C																																										
고객정보를 조회한다	R																																													
4. 모든 단위 프로세스는 하나 이상의 엔티티 타입에 표기가 되었는가?	<table><tr><th>단위프로세스 \ 엔티티타입</th><th>고객</th><th>주문</th><th>주문목록</th><th>제품</th></tr><tr><td>신규고객을 등록한다</td><td>C</td><td></td><td></td><td></td></tr><tr><td>주문을 신청한다</td><td>R</td><td>C</td><td>C</td><td>R</td></tr><tr><td>주문량을 변경한다</td><td></td><td>R</td><td>U</td><td></td></tr><tr><td>주문을 취소한다</td><td></td><td>D</td><td>D</td><td></td></tr><tr><td>제품을 등록한다</td><td></td><td></td><td></td><td>C</td></tr><tr><td>고객정보를 조회한다</td><td>R</td><td></td><td></td><td></td></tr><tr><td>납품업체 정보를 등록한다</td><td></td><td></td><td></td><td></td></tr><tr><td>납품업체 정보를 조회한다</td><td></td><td></td><td></td><td></td></tr></table> <p>주문목록은 생성하고, 수정하고, 삭제는 하지만 데이터가 읽혀지지 않는다.</p>	단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품	신규고객을 등록한다	C				주문을 신청한다	R	C	C	R	주문량을 변경한다		R	U		주문을 취소한다		D	D		제품을 등록한다				C	고객정보를 조회한다	R				납품업체 정보를 등록한다					납품업체 정보를 조회한다				
단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품																																										
신규고객을 등록한다	C																																													
주문을 신청한다	R	C	C	R																																										
주문량을 변경한다		R	U																																											
주문을 취소한다		D	D																																											
제품을 등록한다				C																																										
고객정보를 조회한다	R																																													
납품업체 정보를 등록한다																																														
납품업체 정보를 조회한다																																														
5. 두 개 이상의 단위 프로세스가 하나의 엔티티 타입을 생성하는가?	<table><tr><th>단위프로세스 \ 엔티티타입</th><th>고객</th><th>주문</th><th>주문목록</th><th>제품</th></tr><tr><td>신규고객을 등록한다</td><td>C</td><td></td><td></td><td></td></tr><tr><td>주문을 신청한다</td><td>R</td><td>C</td><td>C</td><td>C</td></tr><tr><td>주문량을 변경한다</td><td></td><td>R</td><td>U</td><td></td></tr><tr><td>주문을 취소한다</td><td></td><td>D</td><td>D</td><td></td></tr><tr><td>제품을 등록한다</td><td></td><td></td><td></td><td>C</td></tr><tr><td>고객정보를 조회한다</td><td>R</td><td></td><td></td><td></td></tr></table> <p>주문목록은 생성하고, 수정하고, 삭제는 하지만 데이터가 읽혀지지 않는다.</p>	단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품	신규고객을 등록한다	C				주문을 신청한다	R	C	C	C	주문량을 변경한다		R	U		주문을 취소한다		D	D		제품을 등록한다				C	고객정보를 조회한다	R													
단위프로세스 \ 엔티티타입	고객	주문	주문목록	제품																																										
신규고객을 등록한다	C																																													
주문을 신청한다	R	C	C	C																																										
주문량을 변경한다		R	U																																											
주문을 취소한다		D	D																																											
제품을 등록한다				C																																										
고객정보를 조회한다	R																																													

"끝"

08	인공지능 신뢰성		
문제	인공지능 신뢰성의 개념과 핵심 속성에 대하여 설명하시오.		
도메인	AI	난이도	상 (상/중/하)
키워드	안전성, 설명가능성, 투명성, 견고성, 공정성, 다양성		
출제배경	인공지능 신뢰성 제도(CAT) 출현에 따른 지식 점검		
참고문헌	ITPE 기술사회 자료 [TTA] 2023 신뢰할 수 있는 인공지능 개발 안내서 - 일반 분야_(ISBN)		
해설자	NS반 김민재 기술사(제 124회 정보관리기술사 / kmj_pe@naver.com)		

I. 인공지능 활용 및 확산 부작용 방지, 인공지능 신뢰성 개념

가. 인공지능 신뢰성 개념

- 데이터 및 모델의 편향, 인공지능 기술에 내재한 위험과 한계를 해결하고, 인공지능을 활용하고 확산하는 과정에서 부작용을 방지하기 위해 준수해야 하는 가치 기준

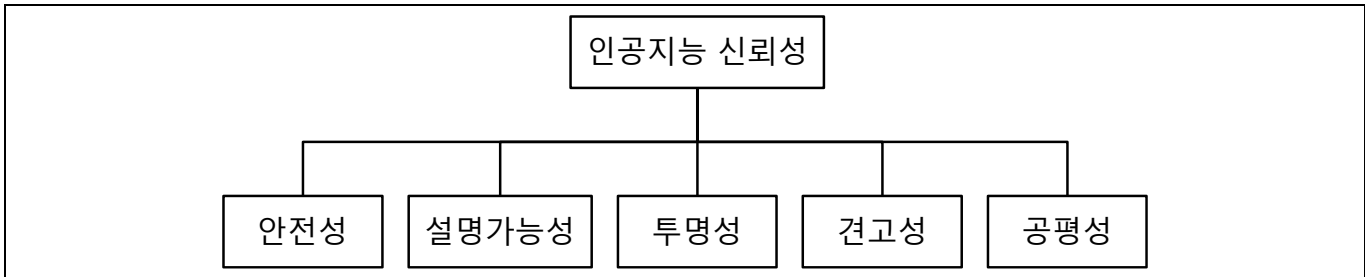
나. 인공지능 신뢰성의 필요성

필요성	설명
윤리적 이슈	- 인공지능을 통해 가짜 사진을 제작·유포하여 사회적 혼란 유발 - 인공지능이 사람에게 윤리적으로 논란이 될 수 있는 행동 권유
안전 사고	- 인공지능과 연동된 하드웨어로 인한 안전사고 발생
설명 가능성	- 인공지능의 추론 결과에 대한 사용자의 문제 제기

- 윤리, 안전, 설명 가능성에 대한 필요성이 증대하여 인공지능 신뢰성이 대두됨

II. 인공지능 신뢰성의 주요 핵심 속성

가. 인공지능 신뢰성의 핵심 속성 분류도



- 인공지능 신뢰성은 5가지 핵심 속성으로 구성

나. 인공지능 신뢰성 핵심 속성 설명

핵심속성	설명
안전성 (safety)	- 인공지능이 판단·예측한 결과로 시스템이 동작하거나 기능이 수행됐을 때 사람과 환경에 위험을 줄 가능성이 완화 또는 제거된 상태

설명가능성 (explainability)	- 인공지능의 판단·예측의 근거와 결과에 이르는 과정이 사람이 이해할 수 있는 방식으로 제시되거나, 문제 발생 시 문제에 이르게 한 원인을 추적할 수 있는 상태
투명성 (transparency)	- 인공지능이 내리는 결정에 대한 이유가 설명 가능하거나 근거가 추적 가능하고, 인공지능의 목적과 한계에 대한 정보가 적합한 방식으로 사용자에게 전달되는 상태
견고성 (robustness)	- 인공지능이 외부의 간섭이나 극한적인 극한적인 운영 환경 등에서도 사용자가 의도한 수준의 성능 및 기능을 유지하는 상태
공평성 (fairness)	- 인공지능이 데이터를 처리하는 과정에서 특정 그룹에 대한 차별이나 편향성을 나타내거나, 차별 및 편향 포함한 결론에 이르지 않는 상태

- 인공지능 신뢰성 속성에 프라이버시, 지속가능성도 핵심 속성 중 하나로서 다양하게 논의 중

### III. 인공지능 신뢰성 요건

구분	설명
다양성 존중	<ul style="list-style-type: none"> <li>- 인공지능이 특정 개인이나 그룹에 대한 차별적이고 편향된 관행을 학습하거나 결과를 출력하지 않으며, 인종·성별·연령 등과 같은 특성과 관계없이 모든 사람이 평등하게 인공지능 기술의 혜택을 받을 수 있는 것</li> <li>- 관련 속성: 공평성·공정성(fairness), 정당성(justice)</li> <li>- 관련 키워드: 편향, 차별, 편견, 다양성, 평등</li> <li>- 국제표준(ISO/IEC TR 24027:2021 - Bias in AI systems and AI aided decision making)에서는 공평성 미정의(공평성은 복잡하고 문화·세대·지역 및 정치적 견해에 따라 다양하여 사회적으로나 윤리적으로 일관되게 정의하기 힘들기 때문)</li> </ul>
책임성	<ul style="list-style-type: none"> <li>- 인공지능이 생명주기 전반에 걸쳐 추론 결과에 대한 책임을 보장하기 위한 메커니즘이 마련되어 있는 것</li> <li>- 관련 속성: 책무성(responsibility), 감사가능성(auditability), 답변가능성(answerability)</li> <li>- 관련 키워드: 책임</li> <li>- 국제표준(ISO/IEC TR 24028:2020 - Overview of trustworthiness in artificial intelligence)에서의 정의: 엔터티의 작업이 해당 엔터티에 대해 고유하게 추적될 수 있도록 하는 속성</li> </ul>
안전성	<ul style="list-style-type: none"> <li>- 인공지능이 인간의 생명·건강·재산 또는 환경을 해치지 않으며, 공격 및 보안 위협 등 다양한 위험에 대한 관리 대책이 마련되어 있는 것</li> <li>- 관련 속성: 보안성, 강건성·견고성, 성능보장성, 통제가능성·제어가능성</li> <li>- 관련 키워드: 적대적 공격(adversarial attack), 복원력(resilience), 프라이버시(privacy)</li> <li>- 국제표준(ISO/IEC TR 24028:2020)에서의 정의: 용인할 수 없는 위험(risk)으로부터의 자유</li> </ul>
투명성	<ul style="list-style-type: none"> <li>- 인공지능이 추론한 결과를 인간이 이해하고 추적할 수 있으며, 인공지능이 추론한 결과임을 알 수 있는 것</li> <li>- 관련 속성: 설명가능성, 이해가능성, 추적가능성, 해석가능</li> <li>- 관련 키워드: 설명가능한 인공지능(XAI, explainable AI), 이해도(comprehensibility)</li> <li>- 국제표준(ISO/IEC TR 29119-11:2020 - Guidelines on the testing of AI-based systems)에서의 정의: 시스템에 대한 적절한 정보가 관련 이해 관계자에게 제공되는 시스템의 속성</li> </ul>

- 인공지능 신뢰성 속성에 대한 충분한 이해를 바탕으로 인공지능 서비스를 제공하는 것이 중요하며, 해당 인공지능 서비스가 고려한 속성에 대해 적절하게 이행하는지 지속 검토 필요

## IV. 주요국 인공지능 신뢰성 관련 정책 동향

국가	주요 정책 (연도)	특징
한국	<ul style="list-style-type: none"> <li>· 사람이 중심이 되는 인공지능을 위한 ('21)</li> <li>· 사람이 중심이 되는 ('20) · ('19)</li> </ul>	<ul style="list-style-type: none"> <li>- 사람 중심 인공지능을 기본가치로 인공지능 생태계 구축·인재양성·산업확산·역기능방지 등 종합 정책 추진</li> </ul>
유럽위원회	<ul style="list-style-type: none"> <li>· 인공지능 규제안('21)</li> <li>· 신뢰할 수 있는 인공지능 윤리 가이드라인 ('19)</li> </ul>	<ul style="list-style-type: none"> <li>- 인간중심의 가치, 윤리, 보안 등 균형 잡힌 인공지능 정책 추진 지향</li> </ul>
유네스코	<ul style="list-style-type: none"> <li>· 인공지능 윤리 권고('21)</li> </ul>	<ul style="list-style-type: none"> <li>- 유네스코 회원국 193개국이 만장일치로 채택한 인공지능 윤리 국제 지침</li> </ul>
미국	<ul style="list-style-type: none"> <li>· 인공지능 권리장전('22)</li> <li>· 인공지능 애플리케이션 규제에 관한 가이드 ('20)</li> </ul>	<ul style="list-style-type: none"> <li>- 산업 분야별 인공지능 활용·촉진을 위한 인공지능 기술 개발 지원과 규제 완화 정책에 중점</li> </ul>
중국	<ul style="list-style-type: none"> <li>· 차세대 인공지능 발전계획('17)</li> </ul>	<ul style="list-style-type: none"> <li>- 정부 주도의 대규모 투자와 강력한 인력양성, 데이터 개방·공유 등 기업친화적 정책 추진</li> </ul>
일본	<ul style="list-style-type: none"> <li>· 인공지능 원칙 구현을 위한 거버넌스 가이드라인('22)</li> <li>· 인공지능 활용전략('19)</li> <li>· 인간중심의 인공지능 사회 원칙('18)</li> </ul>	<ul style="list-style-type: none"> <li>- 경제, 산업, 사회, 윤리 등의 관점에서 포괄적 접근</li> </ul>
싱가포르	<ul style="list-style-type: none"> <li>· 인공지능 거버넌스 프레임워크('20)</li> </ul>	<ul style="list-style-type: none"> <li>- 설명가능성·투명성·공정성, 인간 중심의 원칙에 기반하는 행동지침을 4가지 핵심영역별로 제시</li> </ul>

“끝”



09	BCP(Business Continuity Planning)		
문제	BCP(Business Continuity Planning) 수립 시의 주요 지표와 DRS(Disaster Recovery System) 구축 시의 핵심 고려사항에 대하여 설명하시오.		
도메인	경영전략	난이도	하(상/중/하)
키워드	무중단, BIA, RA, RTO, RPO, Mirror, Hot, Warm, Cold		
출제배경	코로나 사태 및 대기업 데이터 센터 화재 복구 지연등 사회적 이슈에 따른 토픽 확인		
참고문헌	ITPE 모의고사 해설지, ITPE 서브노트		
해설자	강복심화 이제이 기술사(제130회 정보관리기술사/bwmslove@naver.com)		

I. 지속 경영을 위한 핵심 체계, BCP(Business Continuity Planning)

가. BCP의 정의

- 기업 비즈니스에 갑작스런 재해가 발생하더라도 비즈니스를 중단 없이 지속적으로 수행할 수 있도록 IT infra, 인적 자원, 물적 자원에 대한 재해 예방 및 복구 계획을 수립하는 체계

나. BCP의 주요 지표

주요 지표	개념도	설명
RPO (Recovery Point Objective)	<p>복구목표시점(RPO)</p>	<ul style="list-style-type: none"> <li>- 재해로 인하여 중단된 서비스를 복구하였을 때, 유실을 감내할 수 있는 데이터의 손실 허용 시점</li> <li>- 고려된 RTO는 구축비용에 반비례하여 재해 시 발생손실에 비례</li> </ul>
RTO (Recovery Time Objective)	<p>복구목표시간(RTO)</p>	<ul style="list-style-type: none"> <li>- 재해로 인하여 서비스가 중단되었을 때, 서비스를 복구하는 데까지 필요한 최대 허용 시간</li> <li>- 특정 백업 시점(단일 마감, 재해 발생 시점)</li> </ul>

- 이외에 RSO(Recovery Scope Objective), RCO(Recovery Communication Objective), BCO(backup Center Objective)를 고려하여 DRS 구축



## II. DRS(Disaster Recovery System) 개념 및 구축 시 핵심 고려사항

### 가. DRS의 개념

- 정보시스템에 대한 비상 대비체계 유지와 각 업무 조직별 비상사태에 대비한 복구계획 수립을 통한 업무 연속성을 유지할 수 있는 체계

### 나. DRS 구축 시 핵심 고려사항

구분	유형	설명
비즈니스 측면	- 비용성	- 구축비용을 고려하여 기업의 목표와 환경에 맞는 방법 선택
	- 가치증대성	- 기업의 가치를 높일 수 있는 방안 마련
	- 안정성	- BIA, RA를 통한 우선순위 도출
시스템 측면	- 상호운영성	- 기존시스템과의 호환성 고려
	- 복구용이성	- 복구시간, 복구효율성을 높이기 위한 방안 마련
	- 확장성	- 향후 시스템 확장을 고려한 DRS 구축

- DRS 구축 유형은 경제성과 안정성 확보 수준에 따라 결정

## III. RTO와 RPO에 따른 DR 유형 결정 방법

유형	RTO	RPO	비고
Mirror Site	0(이론적)	0	WAS, 높은 비용
Hot Site	< 수시간	≒ 0	데이터 실시간 미러링
Warm Site	수일 ~ 수주	약 수시간 ~ 1일	중요성이 높은 정보기술 자원
Cold Site	수주 ~ 수개월	수일 ~ 수주	데이터만 원격지 보관

- 시스템 및 데이터 중요도, 우선순위 판단하여 DR Site 유형 결정

“끝”

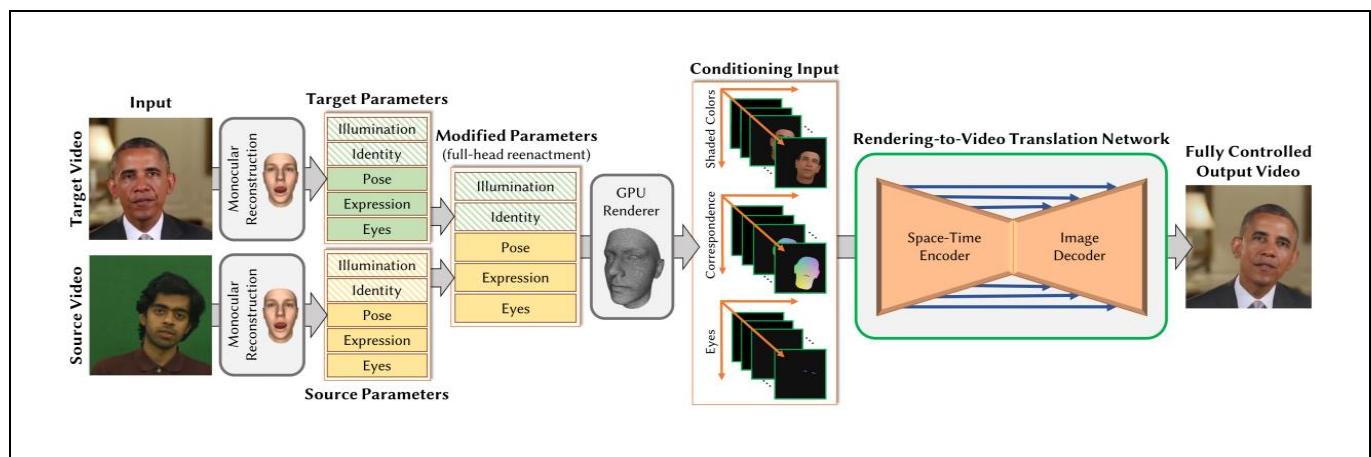
10	딥페이크		
문제	딥페이크(Deepfake)에 대하여 설명하시오.		
도메인	AI	난이도	중(상/중/하)
키워드	GAN, LSTM, 프레임 적용, 이미지 합성		
출제배경	딥페이크를 활용한 가짜영상 악용사례(디지털 성범죄) 증가와 방지법안 법제화에 따른 기술이해		
참고문헌	빅데이터로 본 딥페이크 가짜와의 전쟁 (소프트웨어정책연구소, 2020. 2.25) <a href="https://arxiv.org/pdf/1805.11714.pdf">https://arxiv.org/pdf/1805.11714.pdf</a>		
출제자	강복심화 이제이 기술사(제130회 정보관리기술사/bwmslove@naver.com)		

## I. 진짜 같은 가짜 동영상, 딥페이크(Deep Fake)의 정의

- GAN(Generative Adversarial Network)을 이용해 원본 이미지나 동영상 위에 다른 영상을 중첩하거나 결합하여 원본과는 다른 가공 콘텐츠를 생성하는 합성기술

## II. 딥페이크(Deep Fake)의 메커니즘 및 주요기술

### 가. 딥페이크(Deep Fake)의 메커니즘



- 이미지 영상 수집 -> 이미지 영상 학습 -> Target 영상에 프레임 적용 -> 이미지 합성의 단계를 거쳐 제작

### 나. 딥페이크(Deep Fake)의 주요기술

구분	주요기술	설명
데이터	수집 & 훈련	- Source Video와 Target Video 이미지 및 영상정보 수집하여 GAN 기술을 이용해서 가짜영상을 진짜영상 수준으로 학습
	프레임 적용 & 합성	- 색깔, 형태, 동작 등을 활용하여 Target Video에 프레임 단위로 적용 후 LSTM 네트워크를 활용하여 각 출력 비디오 영상 재생성
알고리즘	GAN	- 동영상의 모션 패턴을 모델링하거나 이미지에서 물체의 3D 모델을 재구성 처리
	LSTM	- 현재 프레임과 이전 프레임의 차를 표현하는 residual 프레임을 CNN과 LSTM 기반의 RNN을 이용하여 벡터화 처리

- 최근 딥페이크를 활용하여 해킹공격에 사용하거나, 가짜뉴스에 악용하는 AI의 역기능 이슈 존재

### III. 딥페이크 이슈 및 진화 방향

이슈 구분	현재	미래
대상	- 사람(유명인)	- 사람(유명인+일반인), 사물, 동물 등
적용 분야	- 성인물, 정치	- 게임, Entertainment 등 다변화
플랫폼	- 독립제작 및 배포	- 독립제작+SNS 결합
규제강도	- 자율규제, 가이드라인	- 입법을 통한 규제 강화

- 최근 온라인상에 범람하고 있는 딥페이크 음란 영상물의 제작·유통을 처벌하는 내용을 담은 딥페이크 처벌법(성폭력처벌법 제14조의2) 통과 (2020. 3. 5)

“끝”

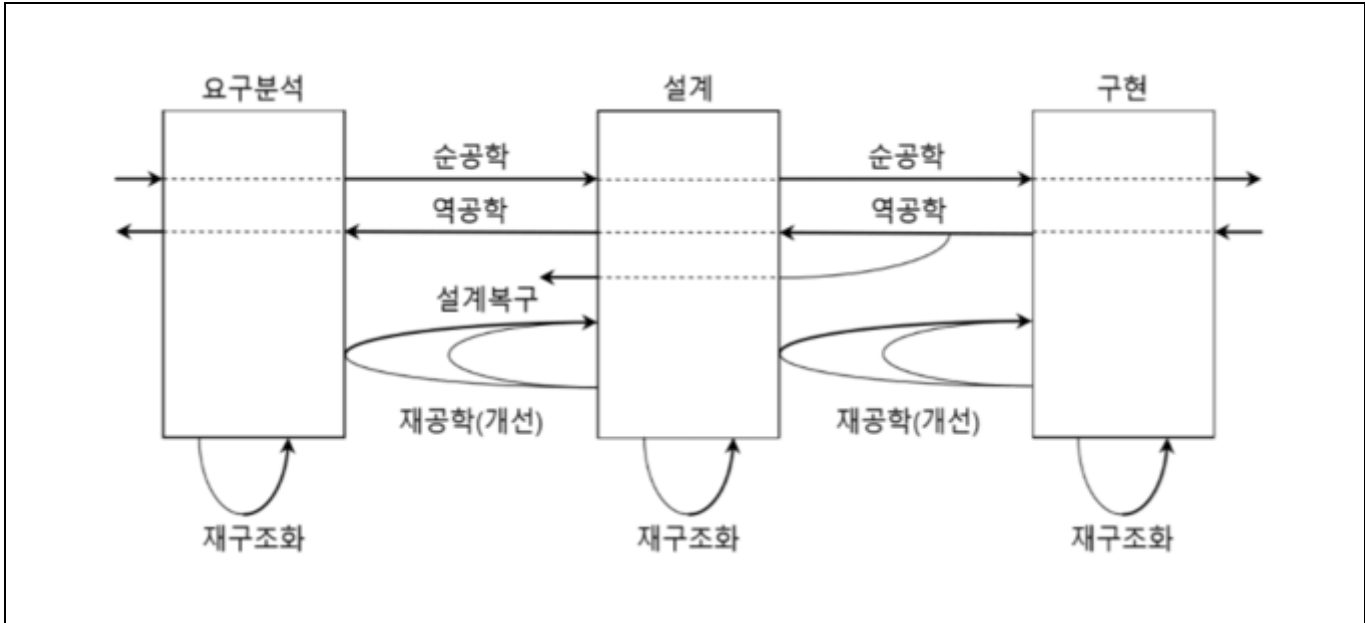
11	3R(Reverse Engineering, Re-Engineering, Reuse)		
문제	소프트웨어 유지보수 향상 및 비용절감을 위한 3R을 설명하시오.		
도메인	SW 공학	난이도	하(상/중/하)
키워드	Reverse Engineering, Reengineering, Reuse, 유지보수 향상, 비용절감		
출제배경	SW 생산성 향상을 위한 정보공학 방법론 확인		
참고문헌	ITPE 기술사회		
해설자	강복심화 이제이 기술사(제130회 정보관리기술사/bwmslove@naver.com)		

I. SW 유지보수 향상 및 비용절감, 3R의 개요

- SW 위기를 극복하고 소프트웨어 생산성을 극대화하기 위해 레포지토리(Repository)를 기반으로 역공학(Reverse Engineering), 재공학(Reengineering), 재사용(Reuse) 기법을 사용하는 공학적 접근법

II. 3R의 개념도 및 상세설명

가. 3R의 개념도



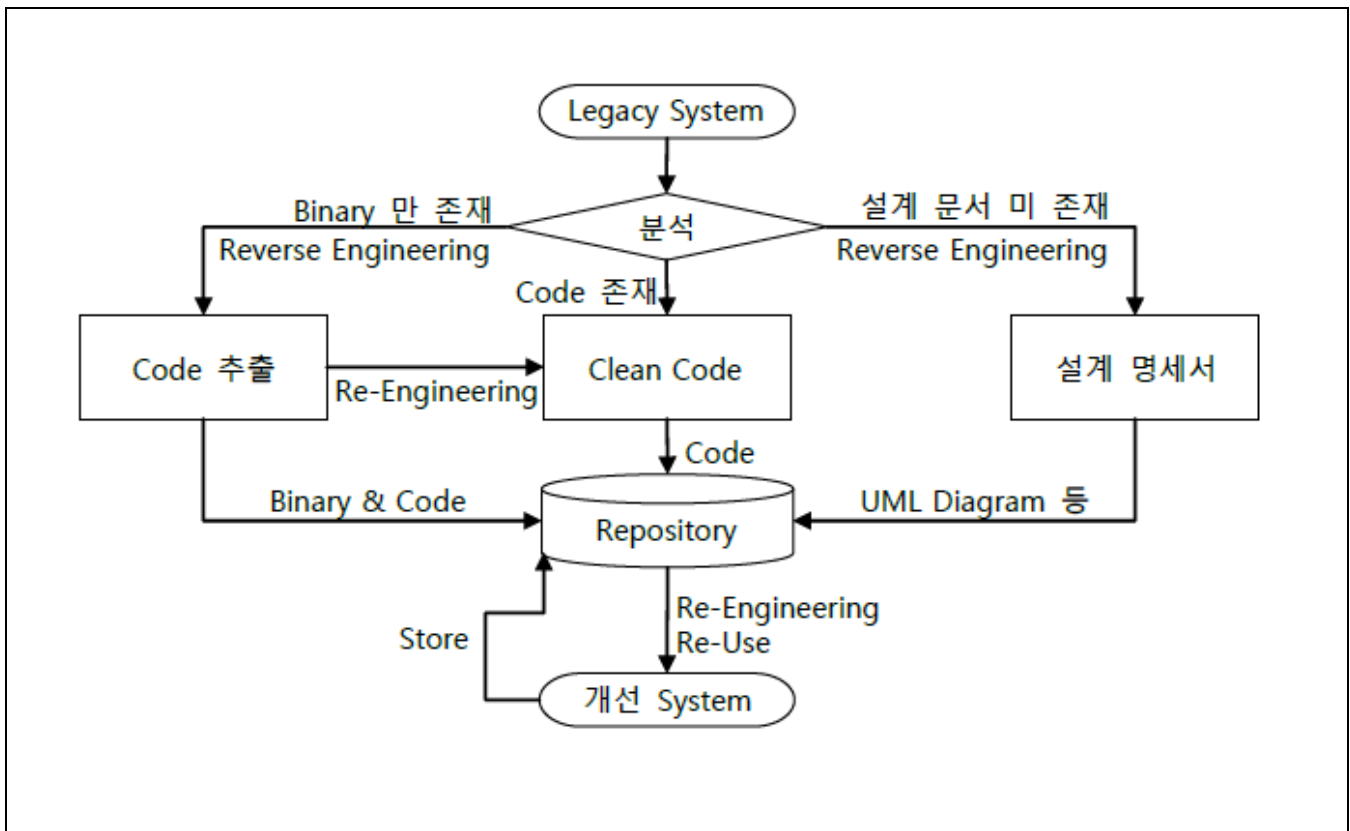
나. 3R의 상세설명

구분	절차	세부 내용
역공학 (Reverse-Engineering)	① Code 추출	- Ollydbg, JD-GUI 등의 Tool 사용 - Dirty Code 추출
	② Code 분석 /수정	- 정적/동적 분석 및 Clean Code 화 - WireShark 와 같은 툴을 사용하여 실제 동작 시 발생하는 Packet, Log file을 추출하여 분석
	③ 문서화	- 분석된 Source code와 해당 분야의 도메인 지식을 활용하여 Program, Data 구조에 대한 명세서 작성

재공학 (Re-Engineering)	① Reverse Engineering	<ul style="list-style-type: none"> <li>- Source Code, 설계 문서 추출</li> <li>- Target System 실행 file의 구조(PE)등을 분석하고 Assembly Code를 추출하는 단계</li> </ul>
	② 재구조화	<ul style="list-style-type: none"> <li>- Target System에서 추출한 code와 repository에서 획득한 code &amp; 문서를 기반으로 의미론적 정보 추출과 Program 구조와 데이터를 재구조화 하는 단계</li> </ul>
	③ 구현	<ul style="list-style-type: none"> <li>- 재구조화된 Program과 Data를 합성하여 성능이 개선된 Target System을 Build 하는 단계</li> </ul>
재사용 (Re-use)	① Forward Engineering	<ul style="list-style-type: none"> <li>- OOP, COBA, COM, 형상관리</li> <li>- 초기 개발 시, 단계별, 모듈별로 분할과 추상화를 통해 구현함으로써, 향후 재사용 가능하도록 하는 단계</li> </ul>
	② Re Use	<ul style="list-style-type: none"> <li>- 생성 중심(Generation Based) : Repository에서 재사용 단위를 찾아 발전시키는 기술</li> <li>- 합성중심(Composition Based) : Repository에서 재사용 가능한 모듈을 추출하여 생산성 있게 조립하는 기술</li> </ul>

- 3R을 통해 유지보수의 향상 및 비용절감 가능

### III. 유지보수 관점의 실무적 3R프로세스



- 3R 각각 요소들의 독립적 수행보다는, 유지보수와 생산성 향상을 위한 상호 연계 적용 필요

“끝”

12	쿠버네티스(Kubernetes)		
문제	쿠버네티스(Kubernetes)		
도메인	디지털서비스	난이도	하(상/중/하)
키워드	Master Node, API Server, Scheduler, Controller Manager, Cloud Controller Manager Worker Node, etcd, Pod, Kubelet, Kube-Proxy, 컨테이너 런타임, cAdvisor, kubectl		
출제배경	서비스 오케스트레이션 플랫폼인 쿠버네티스의 개념 숙지, 이해확인		
참고문헌	<a href="https://kubernetes.io/ko/">https://kubernetes.io/ko/</a> ITPE 기술사회 자료		
출제자	강복심화 이제이 기술사(제130회 정보관리기술사/bwmslove@naver.com)		

## I. 운영자동화 및 오케스트레이션을 위한, 쿠버네티스(Kubernetes)의 개요

### 가. 쿠버네티스(Kubernetes)의 개념

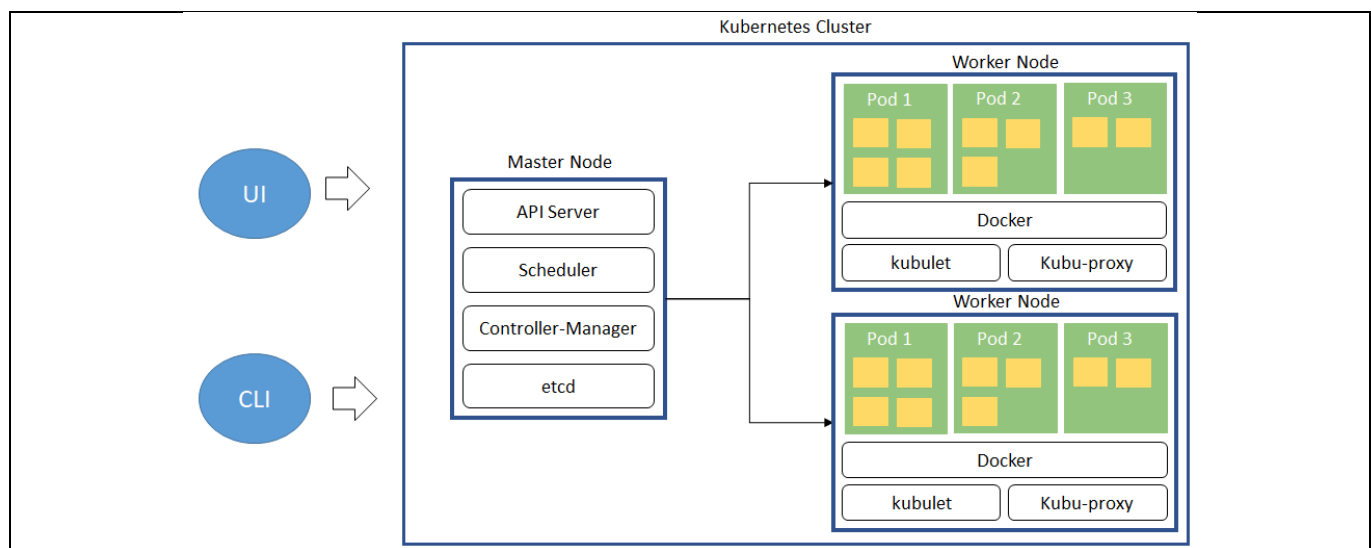
- 컨테이너화된 워크로드와 서비스를 관리하기 위한 이식성과 확장성을 갖춘 오픈소스 오케스트레이션 플랫폼

### 나. 쿠버네티스의 주요 기능

주요기능	설명
복수 호스트 관리	- 여러 호스트에 걸쳐 컨테이너를 오케스트레이션 관리
빈 패킹(bin packing)	- 컨테이너별 필요한 CPU 및 메모리의 크기를 지정
어플리케이션 자동화	- 애플리케이션 배포 및 업데이트를 제어하고 자동화
stateful 어플리케이션	- 스토리지를 장착 및 추가해 스테이트풀(stateful) 애플리케이션을 실행
확장성	- 컨테이너화 된 애플리케이션과 해당 리소스를 즉시 확장 가능
복구 자동화	- 자동 배치, 재시작, 복제, 확장을 사용해 애플리케이션 상태 확인 및 셀프 복구.

## II. 쿠버네티스(Kubernetes)의 아키텍처 및 상세설명

### 가. 쿠버네티스(Kubernetes)의 아키텍처



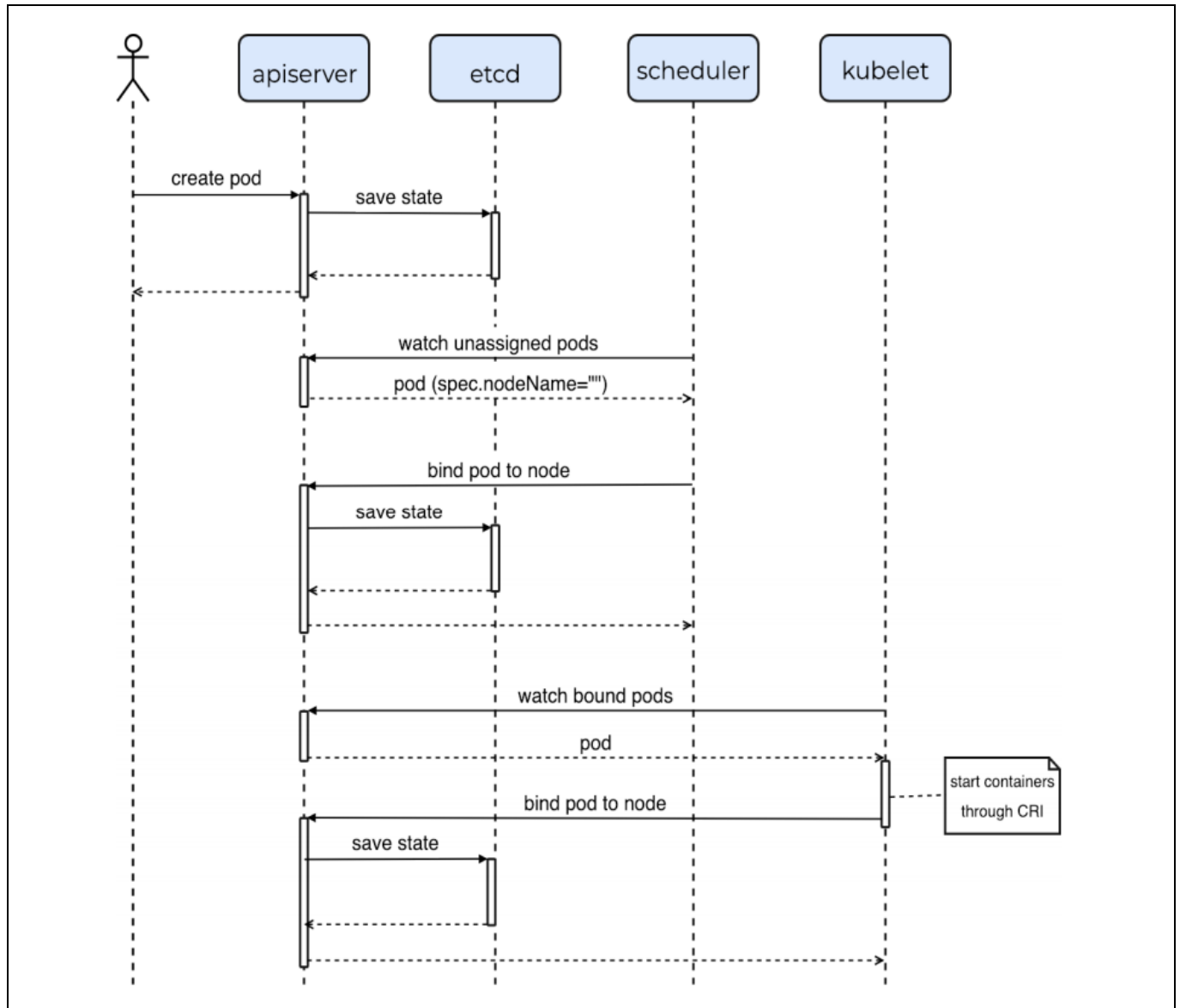
- 마스터 노드를 통해 Worker Node에 있는 POD 를 관리하고, 인증/서비스/복제 등을 제어

### 나. 쿠버네티스(Kubernetes)의 상세설명

구분	구성	설명
Master Node (Control Plane)	Kubu API Server	- 간단하게 명령어를 전달해주는 역할 진행, etcd 클러스터와 REST방식 통신 - kubectl 명령어를 통해 받은 작업을 API서버로 전송
	Kubu Scheduler	- Pod, 서비스 등 각 자원을 적절한 노드에 할당하는 역할 - 공유 상태 스케줄링(Shared-State Scheduling)으로 리소스 할당을 결정
	Kubu Controller Manager	- 여러 Controller들을 생성하고 각 Node에 배포 및 관리 역할
		Replication Controller - Pod 기동 및 장애 발생시 새로운 Pod 생성
		Services Controller - 신규로 생성된 Pod 의 분하 분산 요청 관리
		Node Controller - Node 장애 발생시 알림과 장애 조치 역할 수행
		End-Point Controller - 서비스와 Pod를 연결
	Cloud Controller Manager	- 클라우드 제공자 전용 컨트롤러 (사내 또는 PC내부 환경에서는 제외)
		Node Controller - Node가 클라우드 상에서 삭제되었는지 판별
		Router Controller - 기본 클라우드 인프라에 경로 구성
		Services Controller - 클라우드 사업자 로드밸런서 생성, 업데이트, 삭제
	etcd	- 모든 클러스터 데이터를 담는 저장소
Worker Node	Pod	- 쿠버네티스에서 가장 기본적인 배포 단위로, 컨테이너 포함하는 단위
	Kubelet	- Node에 배포되는 에이전트, Pod에서 컨테이너 동작을 관리 - Master의 API 서버를 통해 명령 수행 및 상태 정보를 Master 로 전달
	Kube-Proxy	- 컨테이너(Pod)간 네트워크 프록시 및 Load Balancing 수행, 간단한 L3 프록시 - 가상 네트워크 상에서 네트워크 트래픽 라우팅 수행 및 서비스와 Pod IP 관리
	컨테이너 런타임	- Pod를 통해서 배포된 컨테이너를 실행. - Docker, containerd, CRI-O, Kubernetes CRI(Container Runtime Interface) 구현
User	kubectl	- 쿠버네티스의 명령 실행 도구
애드온	DNS	- 쿠버네티스 서비스를 위해 DNS레코드를 제공해주는 DNS서버
	Dashboard	- 쿠버네티스 클러스터를 위한 범용의 웹기반 UI
	컨테이너 리소스 모니터링	- 중앙 데이터베이스내의 컨테이너들에 대한 포괄적인 시계열 매트릭스를 기록하고 그 데이터를 열람하기 위한 UI 제공
	클러스터-레벨 로깅	- 검색/열람 인터페이스와 함께 중앙 로그 저장소에 컨테이너 로그를 저장

- 쿠버네티스 클러스터는 컨테이너화된 애플리케이션을 실행하는 노드라고 하는 워커 머신의 집합이며 모든 클러스터는 최소 한 개의 워커 노드 가지고 있음
- 새로운 Pod 생성이 요청되면, kubu scheduler는 Pod을 배치할 노드를 선정하여 Pod을 배치하는데 노드 필터링 과정과 노드 순위 지정 과정, 두 가지 과정을 통해 노드를 선정

### III. 쿠버네티스의 kubernetes scheduler 설명



절차	설명
①	- kubectl apply 등의 명령어를 통해 사용자의 Pod 생성 요청이 kube-apiserver에 제출되면 kube-apiserver는 etcd에 Pod의 정보를 저장
②	- Pod이 위치한 노드의 정보인 nodeName의 값이 설정되지 않은 상태로 Pod의 정보 저장
③	- kube-scheduler는 이러한 정보를 watch 과정을 하고 있다가 Pod의 nodeName이 비어 있는 상태라는 것을 감지
④	- 해당 Pod을 할당하기 위한 적절한 노드를 찾는, 일종의 스케줄링 작업을 진행
⑤	- 적절한 노드를 찾았다면, kube-scheduler는 그 정보를 kube-apiserver에게 전달
⑥	- Pod의 직접 생성은 kubelet에서 처리
⑦	- Pod의 상태 정보를 etcd에 업데이트

- 스케줄링으로 발생하는 모든 상태 정보는 etcd에 상태를 기록/공유하면서 처리

“끝”



13	TCP(Transmission Control Protocol) 프로토콜		
문제	TCP 프로토콜의 3-way handshake와 4-way handshake를 설명하시오.		
도메인	네트워크	난이도	중 (상/중/하)
키워드	신뢰성, 연결지향적, SYN, ACK, FIN		
출제배경	132회 컴퓨터시스템응용기술사 1교시 4-way handshake 기출		
참고문헌	ITPE 기술사회		
출제자	강복심화 이제이 기술사(제130회 정보관리기술사/bwmslove@naver.com)		

I. TCP 전송계층의 개념

구분	설명	
개념	연결지향적이며 신뢰성 있는 데이터 전송을 보장하는 프로토콜	
특징	연결 지향	3-way handshake를 통한 연결확립 후 데이터 전송
	흐름 제어	데이터 처리 속도를 조절하며 수신자의 버퍼 오버플로우 방지
	혼잡 제어	네트워크 내의 패킷 수가 과도하게 증가하지 않도록 방지

- TCP 프로토콜은 연결과정 3-way handshake, 해제과정 4-way handshake 사용

II. 3-way handshake와 4-way handshake 설명

가. 3-way handshake의 설명

구분	설명	
개념도		
설명	초기연결시도	- Client는 접속하고자 하는 서버의 포트번호와 클라이언트의 초기순서번호(Init Sequence Number)를 지정한 SYN 세그먼트를 전송
	서버응답	- 서버의 초기순서번호(ISN)를 포함한 자신의 SYN세그먼트로 응답 - 클라이언트의 ISN + 1 ACK를 보냄으로써 클라이언트의 SYN에 확인 응답
	클라이언트 응답	클라이언트는 서버로부터 보내온 SYN에 대하여 서버의 ISN + 1 ACK로 확인응답을 전송

나. 4-way handshake의 설명

구분	설명	
개념도		
설명	연결종료 요청	- client가 Server에게 연결 종료를 요청
	서버 ACK 신호	- 서버는 바로 종료하지 않고 ACK를 전송해 CLOSE_WAIT 상태로 넘어감
	서버 FIN 신호	- 잔여 작업 종료후 서버는 FIN 신호를 보내고 연결 종료 시도
	클라이언트 ACK	- 클라이언트는 서버의 FIN을 잘 받았다는 ACK를 서버에게 보내고, 클라이언트의 ACK를 받으면 서버는 종료

- TCP는 3-way handshaking을 통해 신뢰성 있는 연결 수행하지만, UDP는 신뢰성 고려하지 않음

III. TCP, UDP, SCTP 비교

비교항목	TCP	UDP	SCTP
프로토콜 ID	17	6	132
관련 표준	RFC 793, 1122	RFC 768, 1122	RFC 2960
신뢰성	신뢰 가능	신뢰 불가능	신뢰 가능
연결지향성	연결 지향	비 연결 지향	연결 지향
흐름제어	제공	없음	제공
혼잡제어	제공	없음	제공
결함허용	아니오	아니오	예 (멀티홈 지원)
데이터 전달	엄격한 순서	무순서	부분적으로 순서화
추가 보안 기능	없음	없음	개선

“끝”

