4장. 데이터베이스와 클라우드 보안

Database & Clould Security

박종혁

서울과학기술대학교 컴퓨터공학과

jhpark1@seoultech.ac.kr

목차

4-1. 데이터베이스 보안의 필요성

1. 데이터베이스 보안의 필요성

- 데이터베이스의 중요한 정보들은 일반적으로 하나의 로컬 시스템에 통합되어 저장

  Ex) 회사 재정 데이터, 기밀 전화 기록, 고객과 직원의 정보, 독점 상품 정보, 의료 정보

- 데이터베이스 정보들은 내·외부 공격자의 오용과 변경의 위협의 대상

- 데이터베이스 보안은 통합된 데이터 관리 전략에 필수적 요소

- 데이터베이스 보안이 데이터베이스 의존의 증가를 따라잡지 못하는 이유 [BENN06]
- 현재의 DBMS의 복잡도와 보안 기술 사이에 심한 불균형
- 보안 기술들이 발전되고 있지만 DBMS의 복잡도 증가는 새로운 취약점과 잠재적 오용을 내포
  1. 데이터베이스는 웹 서비스에서 사용되는 HTTP보다 복잡한 SQL이라는 상호작용 프로토콜을 가짐
     - 효과적인 데이터베이스 보안은 SQL의 보안 취약성에 대한 이해가 필요
  2. 일반적으로 조직에서는 데이터베이스 보안 전담 인력이 부족
     - 대부분의 조직은 데이터베이스 관리자와 보안관리자가 구분
     - 데이터베이스 관리자 - 보안에 대한 제한적 지식 보유하여 보안 기술 적용의 시간 부족
     - 보안 관리자 - DBMS와 데이터베이스에 대한 제한적 지식 보유
  3. 기술이 발달함에 따라 다양한 데이터베이스 플랫폼과 응용프로그램 개발은 보안 실무자 및 인력 양성에 대해 부담을 가중

4-2. 데이터베이스 관리 시스템

2. 데이터베이스 관리 시스템

- 데이터베이스
  - 여러 응용프로그램이 사용하는 구조화되어 저장된 데이터의 집합
  - 데이터
  - 데이터 그룹 간의 관계도 포함

- 데이터 파일과 데이터베이스의 차이
  - 간단한 인사 파일은 개인들의 레코드 집합으로 구성
    - 레코드 - 이름, 주소, 생일, 직위, 급여와 같은 정보
  - 인사 데이터베이스 - 인사 파일을 포함하고 각 직원들의 근무 시간 같은 근무 일지 파일도 포함
  - 데이터베이스에서는 이러한 파일들이 연결되어 급여 지급 프로그램이 근무시간과 직원의 급여 수준 정보를 추출하여 급여를 산출

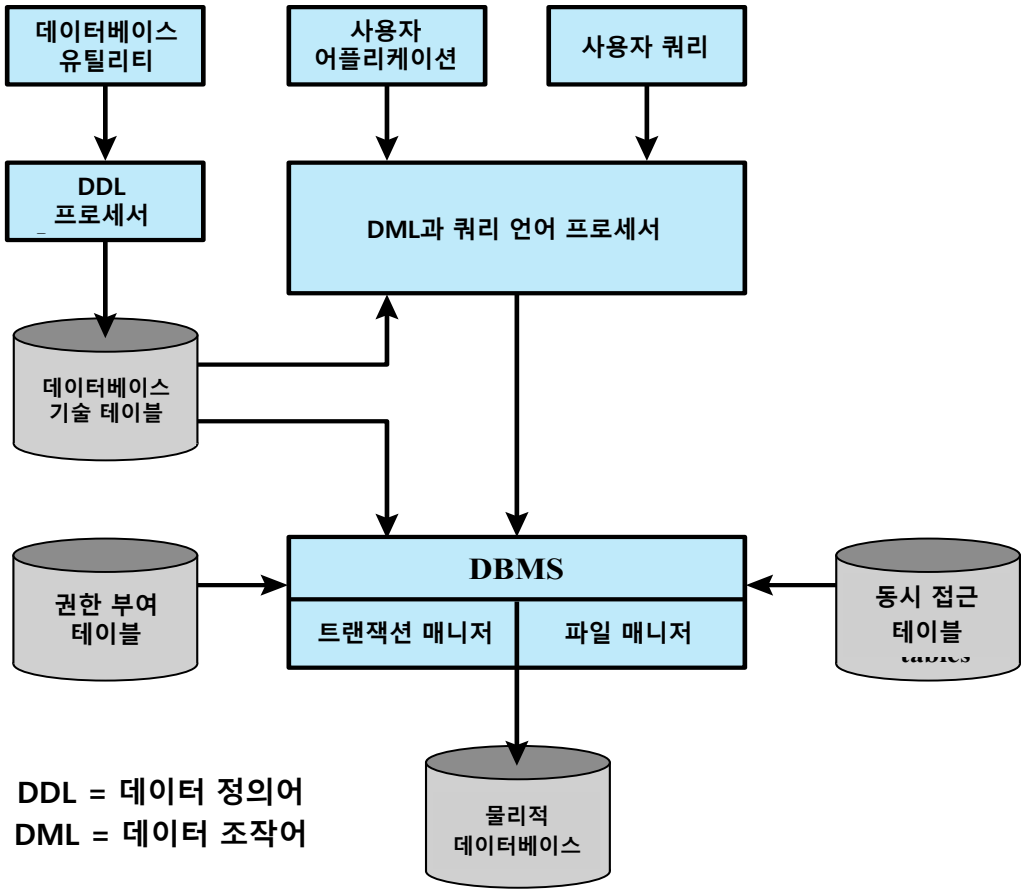
- 데이터베이스 관리 시스템 (DBMS : DataBase Management System)
  - 데이터베이스를 구축 및 유지하면서 다수의 사용자와 응용프로그램들에게 쿼리를 제공하는 프로그램군
  - 데이터베이스 테이블을 사용해 물리적 데이터베이스를 관리
- 데이터베이스 인터페이스
  - 파일 관리 모듈과 트랜잭션 관리 모듈을 통하여 제공
- 권한 테이블
  - 사용자들이 쿼리 언어 명령문을 데이터베이스에 실행하는 권한을 보증
- 동시 접근 테이블
  - 동시에 충돌되는 명령이 실행되었을 때 충돌을 방지

- 데이터 정의어(DDL : Data Definition Language)
  - 데이터베이스의 논리적 구조와 절차적 속성을 정의
  - CREATE, ALTER, DROP, CASECADE, RESTRICTED
- 데이터 조작어(DML : Data Manipulation Language)
  - 응용프로그램 개발자에게 강력한 도구를 제공
  - INSERT, DELETE, UPDATE, SELECT
- 쿼리 언어(Query)
  - 사용자를 지원하기 위한 선언적 언어

- 데이터베이스 시스템은 많은 양의 데이터에 대한 효율적인 접근을 제공

- 복잡도와 중요성 때문에 데이터베이스 시스템은 보편적인 OS 기반 보안 메커니즘이나 독립형 보안 패키지의 능력 이상의 보안을 요구

- DBMS는 상세한 접근 제어를 제공
  - 데이터베이스 내의 특정 항목에 대한 선택, 삽입, 수정, 삭제 같은 넓은 범위의 명령어


데이터베이스
유틸리티
사용자
어플리케이션
사용자 쿼리
DDL
프로세서
DML과 쿼리 언어 프로세서
데이터베이스
기술 테이블
DBMS
권한 부여
테이블
트랜잭션 매니저
파일 매니저
동시 접근
테이블
DDL = 데이터 정의어
DML = 데이터 조작어
물리적
데이터베이스

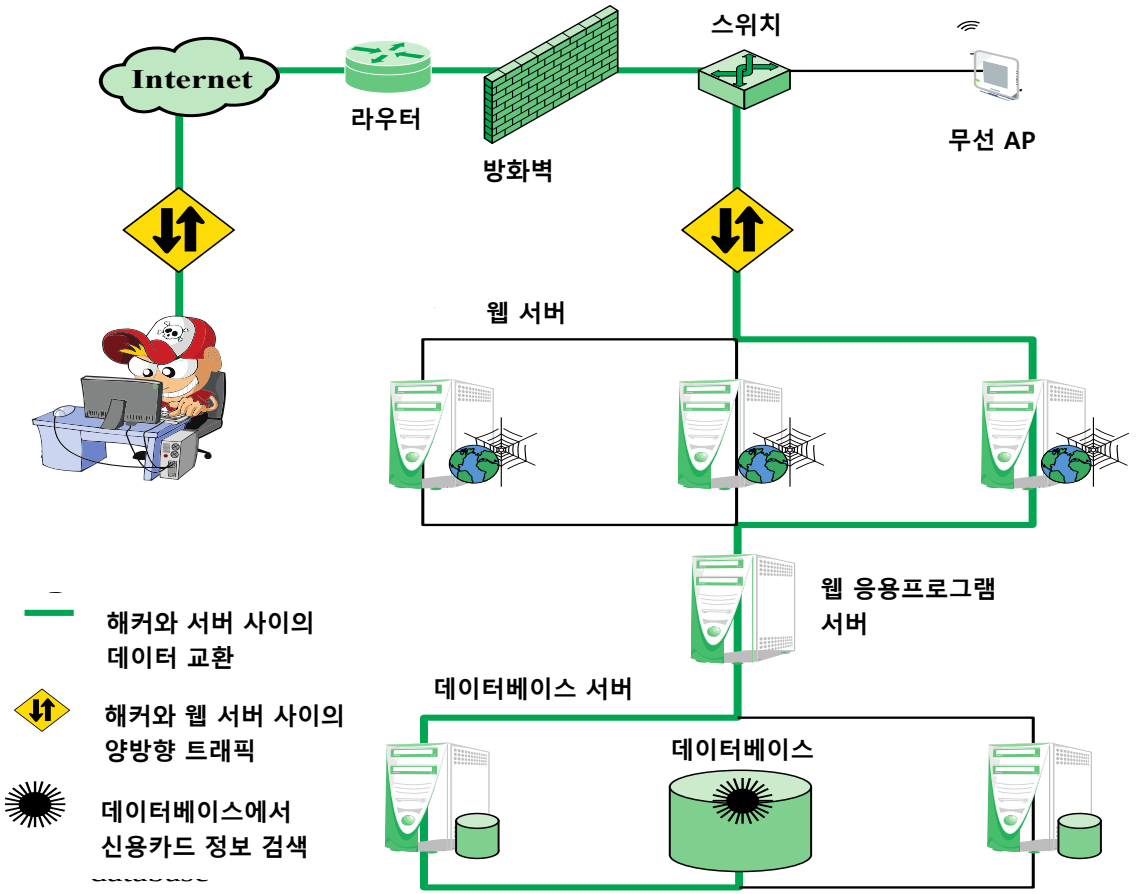
DBMS 구조

4-3. SQL 주입 공격

3. SQL 주입 공격

- SQLi : SQL Injection Attacks
- 가장 널리 사용되며 위협적인 네트워크 기반 보안 공격
- 웹 응용프로그램 페이지의 특성을 활용하도록 설계
- 위해한 SQL 명령을 데이터베이스 서버에 전송
- 가장 흔한 공격 목적: 대량의 데이터 추출
- SQL 주입의 활용
  - 데이터의 삭제, 변경
  - 임의의 OS 명령 수행
  - DoS 공격 실행

- SQL 주입 공격 과정
  1. 공격자는 웹 응용프로그램의 취약성을 찾고 웹 서버에 명령을 보내 SQL 명령을 데이터베이스에 주입
     - 명령은 방화벽이 허용하는 트래픽에 주입
  2. 웹 서버는 위해 코드를 수신하고 웹 응용프로그램 서버로 전송
  3. 웹 응용프로그램 서버는 위해 코드를 웹 서버로부터 수신하고 데이터베이스 서버로 전송
  4. 데이터베이스 서버는 위해 코드를 데이터베이스에 실행
     - 데이터베이스는 신용카드 테이블에서 데이터를 반환
  5. 웹 응용프로그램 서버는 데이터베이스의 신용카드 정보를 포함하는 데이터로 동적으로 페이지를 생성
  6. 웹 서버는 신용카드 정보를 해커에 전송


Internet
라우터
방화벽
스위치
무선 AP
웹 서버
웹 응용프로그램
서버
데이터베이스 서버
데이터베이스
해커와 서버 사이의
데이터 교환
해커와 웹 서버 사이의
양방향 트래픽
데이터베이스에서
신용카드 정보 검색

통상적 SQL 주입공격

- 주입 기법
  - SQLi 공격은 통상적으로 조기 종료된 문자열에 새 명령어를 추가
  - 삽입된 명령은 실행 전에 부가적으로 추가된 문자열을 가질 수 있기 때문에, 공격자는 주입된 문자열을 주석 표기 "- -"로 끝마침
  - 후속 텍스트는 실행 시 무시됨

SQLi 공격 방법

- 사용자 입력
  - 공격자는 적절하게 조합된 사용자 입력을 통해 SQL 명령어를 주입
- 서버 변수
  - 공격자는 HTTP, 네트워크 헤더의 값들을 위조할 수 있으므로, 값들을 직접 헤더 안에 넣을 수 있음
  - 데이터베이스의 서버 변수를 주입하는 쿼리가 발생하면, 위조된 헤더 내의 공격이 동작
- 2차 주입
  - 악의적인 사용자가 시스템이나 데이터베이스에 존재하는 데이터를 이용하여 SQL 주입 공격
  - 공격이 발생하면 공격을 발생하도록 쿼리를 변경한 입력은 사용자가 만든 것이 아니고 시스템이 만든 것임
- 쿠키 변경
  - 공격자는 쿠키를 변경하여 응용프로그램 서버가 변경된 내용, 구조, 기능을 갖는 쿠키를 사용하여 SQL 쿼리를 만들도록 할 수 있음
- 물리적 사용자 입력
  - 웹 요청 범위 밖의 공격을 만드는 사용자 입력을 제공함으로써 가능

SQLi 공격 유형

대역 내 공격 (Inband Attacks)

- SQL 코드 주입과 결과 변환 - 같은 통신 채널을 사용
- 반환된 데이터는 응용프로그램 웹 페이지에 직접 표현
- 공격 유형
  - 동의어 중복: 하나 혹은 여러 조건문에 코드를 주입하여 항상 참이 되도록 함
  - 라인 종단 주석: 특정 필드에 코드를 주입한 후, 뒤의 정상적인 코드는 라인 종단 주석의 사용으로 무효화됨
  - 피기백 (Piggybacked) 쿼리: 공격자는 쿼리를 추가하여, 정상 요청 위에 피기백 쿼리를 편승시킴

- 추론적 공격 (Inferential Attack)
  - 실제 데이터 전송은 없고, 공격자가 특정 요청을 보내고 웹사이트/데이터베이스 서버의 결과 행위를 관찰함으로써 정보를 재구성

    1. 불법/논리적으로 틀린 쿼리
       - 공격자가 후위의 웹 응용프로그램 데이터베이스의 유형과 구조에 대한 정보를 수집할 수 있게 함
       - 공격은 정보-수집의 예비 단계로 간주됨
    2. 블라인드 SQL 주입
       - 공격자에게 시스템 오류 정보를 보여주지 않을 만큼 안전하더라도 데이터베이스 시스템에 존재하는 데이터를 추론할 수 있게 함

- 대역 외 공격 (Out-of-Band Attack)
  - 데이터가 다른 채널로 변환

    예) 쿼리 결과 이메일이 생성되어 시험자에게 보내짐

  - 정보 검색에 제한이 있지만 데이터베이스 서버의 출력 연결이 느슨할 때 사용 가능

SQLi 대응

1. 방어적 코딩
   - 수동 방어적 코딩 습관
     - SQLi 공격이 활용하는 흔한 취약점은 불충분한 입력 실증
     - 이러한 취약성을 없애기 위한 쉬운 솔루션
     - DBMS의 오류를 야기하는 공격을 회피할 수 있음
   - 매개변수 쿼리 삽입
     - 응용프로그램 개발자가 보다 정확히 SQL 쿼리 구조를 명기
     - 독립적으로 매개변수 값을 전달하여 의심되는 사용자 입력이 쿼리 구조를 변경하지 못하게 함
   - SQL DOM
     - 자동화된 데이터 타입 실증과 회피를 가능하게 하는 클래스 집합
     - 데이터베이스 쿼리를 캡슐화하여 안전하고 신뢰성 있게 데이터베이스를 접근
     - 규제받지 않는 문자열 연결 쿼리 생성 과정을 타입 확인 API를 사용하는 체계적인 것으로 바꿈

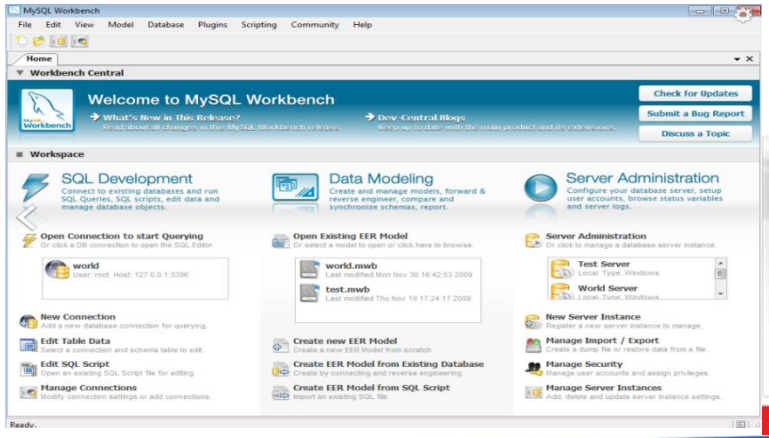
2. 탐지

- 시그니처 기반 (Signature based)
  - 특정 공격 패턴을 찾음
  - 항상 업데이트 되어야 하며 자체 수정 공격에는 대응이 미비함
- 변칙 기반 (Anomaly based)
  - 정상 행위를 정의하고 정상 범위 밖의 행위 패턴을 탐지
  - 시스템이 정상 행위 범위를 학습하는 훈련 단계와 이후의 실재 탐지 단계가 있음
- 코드 분석 (Code analysis)
  - SQLi 취약성을 탐지하는 시험을 포함
  - 시험은 폭넓은 범위의 SQLi 공격을 생성하고 시스템 응답을 평가하도록 설계됨

3. 실시간 방지

- 실시간으로 쿼리를 확인
- 자동화된 솔루션이 다양하게 존재

- 대표적인 DB 솔루션
  - MySQL : 오픈소스 DBMS시장에서 많이 사용, 현재 오라클과 합병
  - 오라클 : DBMS 시장의 대다수를 차지함
  - MS SQL Server : 마이크로소프트에서 만든 DBMS
  - DB2 : IBM 에서 만든 DBMS
  - PostgreSQL : 오픈소스 DBMS, 현재 MySQL과 함께 오픈소스 관계형 DBMS 시장에서 많이 사용함



4-4. 데이터베이스 접근 제어

4. 데이터베이스 접근 제어

- 사용자가 데이터베이스 접근 권한을 갖는지 결정하는 전체적인 접근 제어 시스템을 사용
- 데이터베이스 사용 권한을 인증받은 사용자에게 데이터베이스의 일부에 접근하는 특정 능력 부여
- 다른 접근 권한, 생성, 삽입, 업데이트, 읽기, 쓰기를 구별
- 데이터베이스 접근 제어 관리 정책
  - 중앙관리(Centralized administration)

    소수의 특권 사용자들이 접근 권한을 부여, 파기
  - 소유권 기반의 관리(Ownership-based administration)

    테이블 소유자(생성자)가 테이블 접근 권한을 부여, 파기
  - 분산적 관리(Decentralized administration)

    테이블 소유자가 다른 사용자에게 테이블 접근 권한을 부여 (grant) 및 파기(revoke) . 그들에게 테이블의 접근권한을 부여 및 파기를 허가

SQL 기반 접근 정의

- 두 가지 접근 권한 관리 명령어

1. Grant
   - 하나 혹은 여러 접근 권한을 허가하거나 사용자 역할을 부여

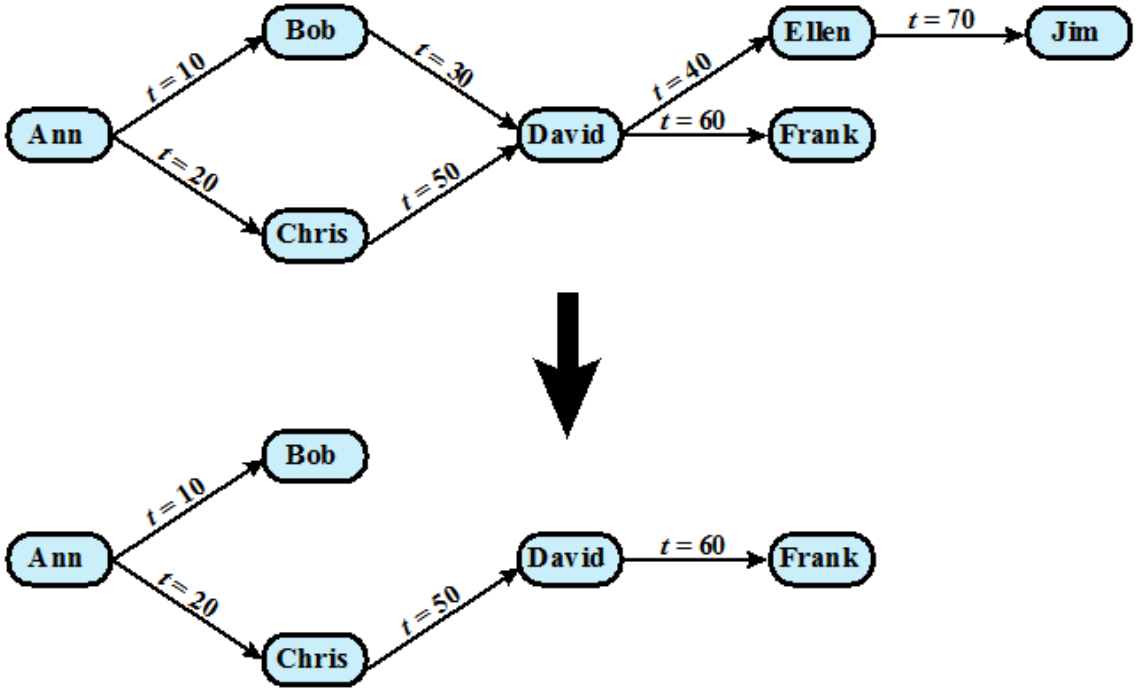
   예) GRANT SELECT ON ANY TABLE TO ricflair

2. Revoke
   - 접근 권한을 파기

   예) REVOKE SELECT ON ANY TABLE FROM ricflair

- 접근 권한 목록
  1. Select
     - 허가자는 전체 데이터베이스, 개개의 테이블 혹은 테이블의 특정 열을 읽기 가능
  2. Insert
     - 허가자는 테이블에 행 삽입
     - 테이블 특정 값을 갖는 열들에 행 삽입
  3. Update
     - 의미론적으로 INSERT와 유사
  4. Delete
     - 허가자는 테이블의 행 삭제 가능
  5. References
     - 허가자는 다른 테이블의 특정 열을 외래키로 정의 가능

- 계단식 권한 부여
  - 권한부여: 한 사용자에서 다른 사용자에게로 계단식으로 주어짐
  - 권한파기: 권한의 파기도 계단식으로 이뤄짐



[Bob이 David로부터 권한을 파기하는 과정]

역할 기반 접근 제어(RBAC)

- 역할 기반 접근 제어(RBAC)는 데이터베이스 접근 제어에 적합

- 데이터베이스 RBAC은 다음 기능이 제공되어야 함
  1. 역할의 생성과 삭제
  2. 역할의 허가 정의
  3. 사용자 역할의 할당 및 취소

- 데이터베이스 사용자 (임의의 접근 제어 환경)
  1. 응용프로그램 소유자
     - 응용프로그램의 일부분으로서 데이터베이스 객체(테이블, 행, 열)를 소유한 사용자
     - 데이터베이스 객체는 응용프로그램에 의해 생성되거나 사용
  2. 응용프로그램 소유자를 제외한 사용자
     - 데이터베이스 객체를 소유하지 않지만 특정 응용프로그램을 통해 데이터베이스 객체를 동작하는 사용자
  3. 관리자
     - 데이터베이스 일부분 혹은 전체에 대한 관리적 책임이 있는 사용자

- 고정된 서버 역할
  - 서버 레벨에서 정의되고 사용자 데이터베이스와 독립적으로 존재
  - 역할은 다른 허가들을 가지며 행정적 책임의 완전한 제어를 포기하지 않으면서 분산시킴
  - 데이터베이스 관리자는 고정된 서버 역할을 여러 사람에게 다른 행정적 업무를 맡기는데 사용할 수 있음

- 고정된 데이터베이스 역할
  - 개별 데이터베이스 레벨에서 동작
  - 고정된 서버역할과 같이 db_accessadmin과 db_securityadmin같은 역할은 DBA의 행정적 책임을 보조하기 위해 설계
  - db_datareader와 db_datawriter는 사용자에게 포괄적 권한을 제공하기 위해 설계

Fixed Roles in Microsoft SQL Server

Role	Permissions
Fixed Server Roles	
sysadmin	Can perform any activity in SQL Server and have complete control over all database functions
serveradmin	Can set server-wide configuration options, shut down the server
setupadmin	Can manage linked servers and startup procedures
securityadmin	Can manage logins and CREATE DATABASE permissions, also read error logs and change passwords
processadmin	Can manage processes running in SQL Server
dbcreator	Can create, alter, and drop databases
diskadmin	Can manage disk files
bulkadmin	Can execute BULK INSERT statements
Fixed Database Roles	
db_owner	Has all permissions in the database
db_accessadmin	Can add or remove user IDs
db_datareader	Can select all data from any user table in the database
db_datawriter	Can modify any data in any user table in the database
db_ddladmin	Can issue all Data Definition Language (DDL) statements
db_securityadmin	Can manage all permissions, object ownerships, roles and role memberships
db_backupoperator	Can issue DBCC, CHECKPOINT, and BACKUP statements
db_denydatareader	Can deny permission to select data in the database
db_denydatawriter	Can deny permission to change data in the database

- 사용자 정의 역할
  - 데이터베이스의 일부에 접근 권한을 지정
  - 적합한 권한을 가진 사용자 또는 관리자는 새로운 역할과 연계된 접근 권한을 정의
  - 사용자 정의 역할
    1. 표준 역할

       인가된 사용자가 다른 사용자에게 역할을 지정
    2. 응용프로그램 역할
       - 사용자 그룹보다는 응용프로그램과 연관되어 있으며 패스워드를 요구
       - 응용프로그램이 적절한 코드를 실행할 때 활성화
       - 응용프로그램에 접근하는 사용자는 데이터베이스 접근에 응용프로그램 역할을 사용

4-5. 추 론 (Inference)

5. 추 론 (Inference)

- 인가된 쿼리 수행과 합법적 응답에서 비인가된 정보를 추론

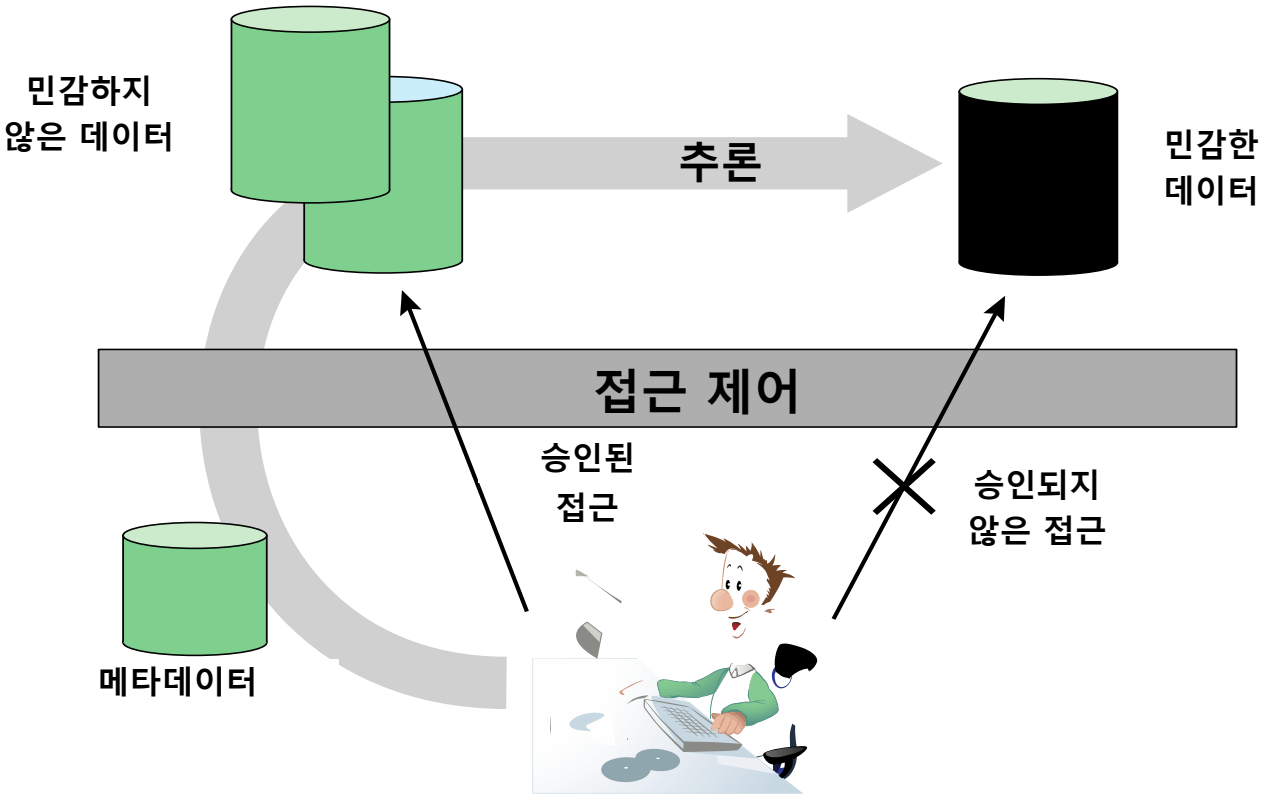
- 추론 문제의 발생
  - 많은 데이터 아이템의 조합이 개별 아이템보다 민감
  - 데이터 아이템들의 조합이 보다 민감한 데이터를 추론

- 공격자는 메타데이터 or 민감하지 않은 데이터를 사용

  ※ 메타데이터(Metadata)

  특정 사용자가 정보를 추론하는 데 사용할 수 있는 데이터 아이템들 간의 상관관계 혹은 의존성에 대한 지식

- 비인가된 데이터의 정보 전송 경로는 추론 채널(inference channel)를 통해서 전송



추론 채널을 통한 간접 정보 접근

- 추론에 의한 노출 위협에 대한 접근 방법
  1. 데이터베이스 설계 시 추론 탐지
     - 데이터베이스 구조 변경 혹은 접근 제어 방식의 변경으로 추론 채널을 제거함으로써 추론을 방지
     - 테이블을 여러 테이블로 분할하여 데이터 의존성을 제거하거나 RBAC 방식에서보다 세밀한 접근 제어 역할을 포함
     - 불필요하게 엄격한 접근 제어때문에 가용성을 제한

  2. 쿼리 타임 시 추론 탐지
     - 쿼리 처리 중 추론 채널 위반을 제거하려는 시도
     - 추론 채널이 탐지되면 쿼리를 거부하거나 변경

추론의 예

- 뷰 사용자들은 항목과 가격의 관계에 접근이 허가되지 않음
  - 항목을 알더라도 가격을 추론할 만한 충분한 정보 같은 기능적 관계가 없음을 의미

- Inventory 테이블의 구조를 알고 Inventory 테이블과 같은 행 순서를 갖는 뷰 테이블들을 아는 사용자는 그림의 테이블을 구성하기 위해 두 뷰를 병합할 수 있음
  - 항목과 비용의 속성들이 노출되지 말아야 하는 접근 제어 정책을 위반

항목	이용 가능한 곳	가격($)	종류
선반 지지대	매장/온라인	7.99	철물
덮개 지지대	온라인	5.49	철물
장식 체인	매장/온라인	104.99	철물
케이크 팬	온라인	12.99	가정용품
샤워/튜브 클리너	매장/온라인	11.99	가정용품
밀 방망이	매장/온라인	10.99	가정용품

(a) 물품 목록 테이블

이용 가능한 곳	가격($)
매장/온라인	7.99
온라인	5.49
매장/온라인	104.99

항목	종류
선반 지지대	철물
덮개 지지대	철물
장식 체인	철물

(b) 2개의 뷰

항목	이용 가능한 곳	가격($)	종류
선반 지지대	매장/온라인	7.99	철물
덮개 지지대	온라인	5.49	철물
장식 체인	매장/온라인	104.99	철물

(c) 쿼리에 대한 답을 혼합한 결과 파생된 테이블

4-6. 데이터베이스 암호화

6. 데이터베이스 암호화

- 데이터베이스는 조직의 가장 가치 있는 정보자원
  - 여러 보안 계층으로 보호
    - 방화벽, 인증 메커니즘, 접근 제어 시스템, 데이터베이스 접근 제어 시스템, 데이터베이스 암호화
    - 암호화는 데이터베이스 보안의 최후의 방어선
  - 전체 데이터베이스에 대해 레코드 레벨(선택된 레코드 암호화), 속성 레벨(선택된 속성 암호화), 혹은 개별 필드 레벨에 대해 적용 될 수 있음

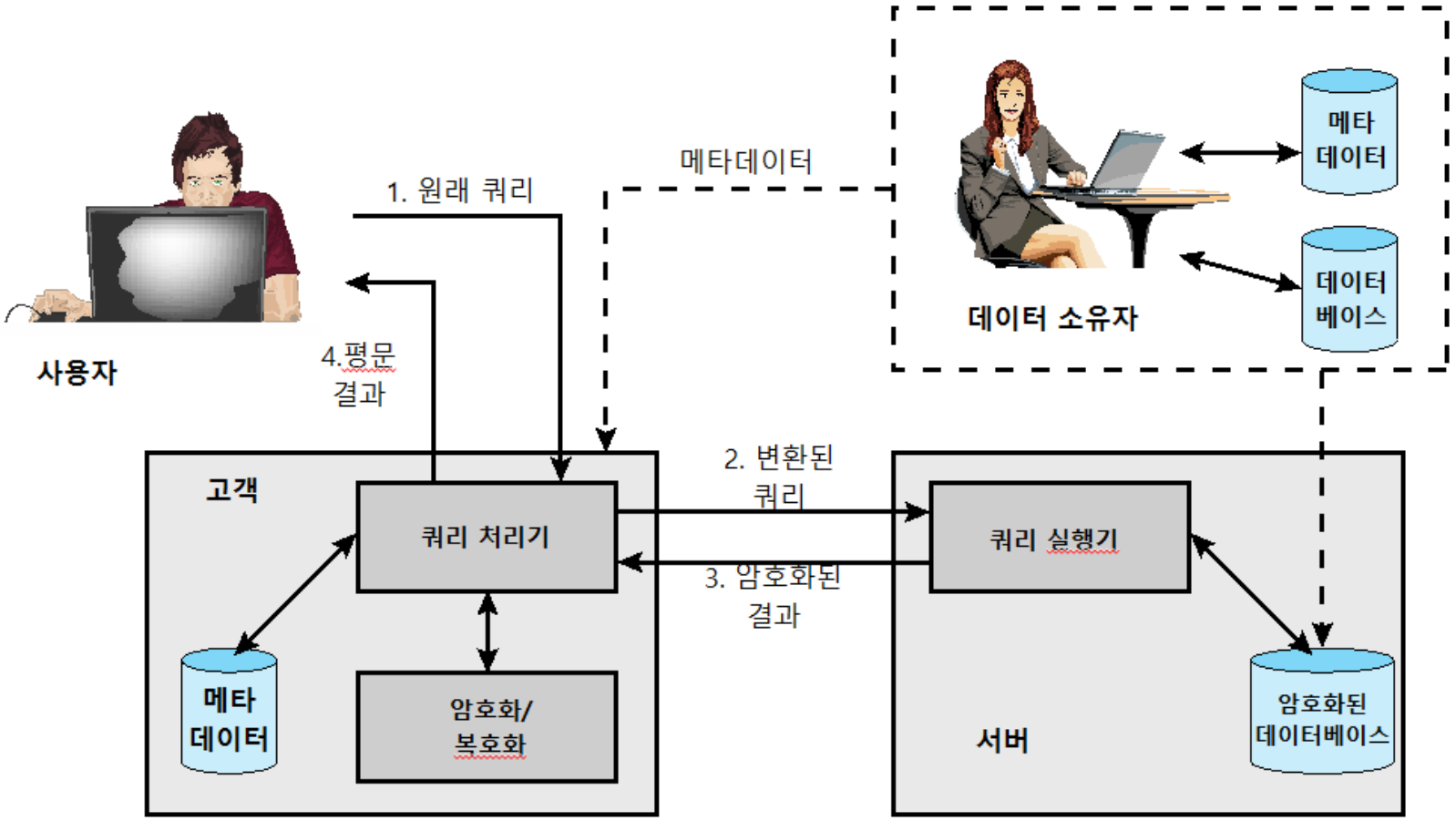
- 데이터베이스 암호화의 단점

1. 키 관리
   - 인가된 사용자는 접근하는 데이터의 복호화 키에 접근해야만 함
   - 일반적으로 폭넓은 사용자와 응용프로그램이 접근 가능
   - 인가된 사용자와 응용프로그램에게 선택된 부분의 데이터베이스의 보안키를 제공하는 것은 복잡함

2. 비유연성
   - 데이터베이스의 부분이나 전체가 암호화되면 레코드 검색이 어려움

데이터베이스 암호화 기법

- 데이터 소유자(Data owner)
  - 조직 내 또는 외부 사용자에게 가능한 데이터를 생성하는 기관
- 사용자(User)
  - 요청(쿼리)을 시스템에게 보내는 사람
  - 사용자는 서버를 통해 데이터베이스에 접근 허가를 받은 조직의 고용인
  - 인증을 통해 접근 허가를 받은 조직 외부의 사용자
- 고객(Client)
  - 사용자 쿼리를 서버에 저장된 암호화된 데이터에 대한 쿼리로 전환하는 객체
- 서버(Server)
  - 데이터 소유자로부터 암호화된 데이터를 받고 고객에게 배포하는 기관
  - 서버는 데이터 소유자가 소유할 수도 있지만, 통상적으로는 외부 제공자에 의해 소유되고 관리됨



데이터베이스 암호화 기법

데이터베이스 암호화 과정

1. 사용자는 특정 기본키 값으로 하나 혹은 다수의 레코드의 필드를 위한 SQL 쿼리를 발행

2. 고객 측의 쿼리 처리기는 기본키를 암호화하고, SQL 쿼리를 수정하여, 서버에 전송

3. 서버는 쿼리를 암호화된 기본키 값으로 처리하고 적절한 레코드를 반환

4. 쿼리 처리기는 데이터를 복호화하고 결과를 반환

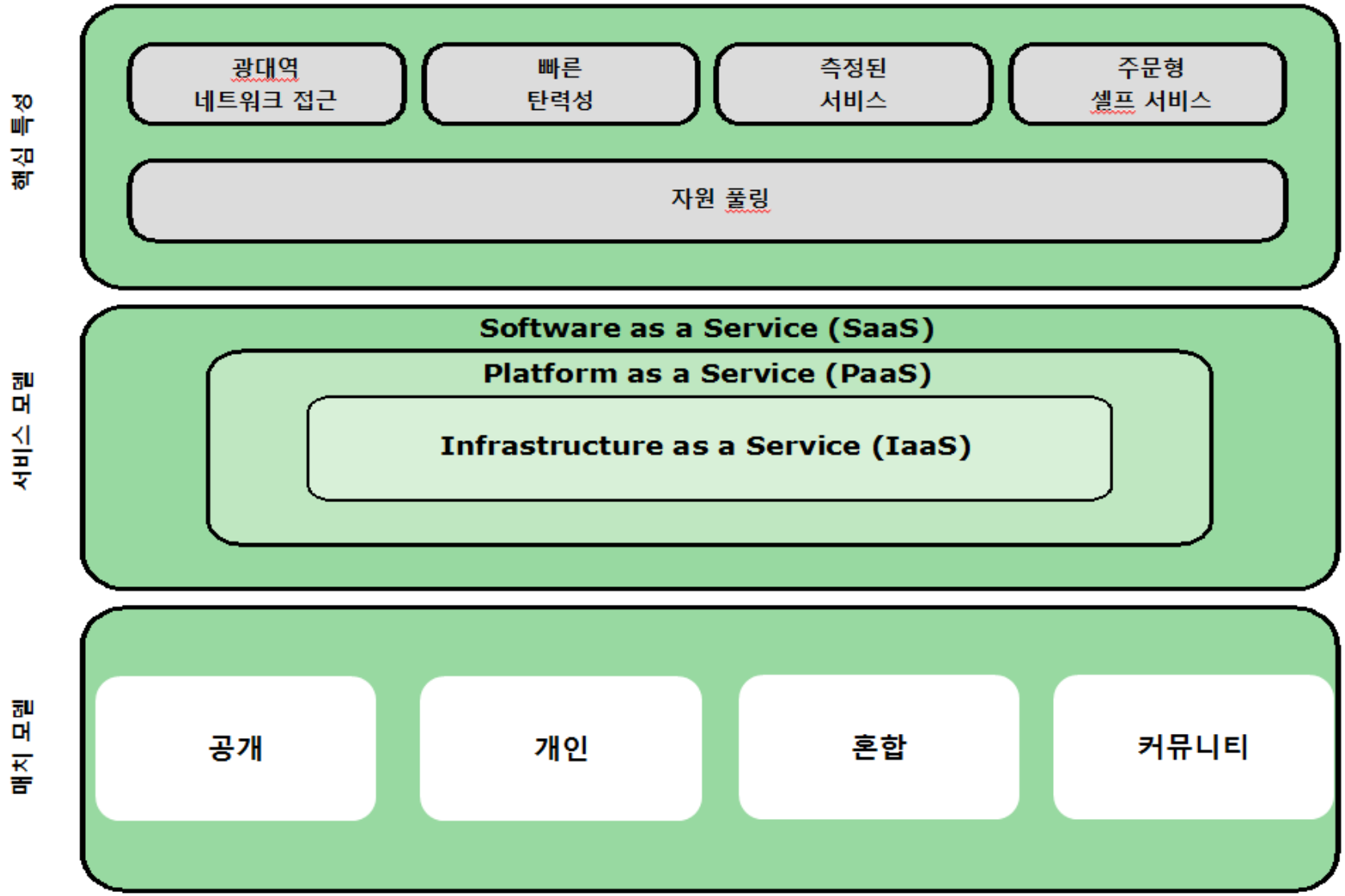
4-7. 클라우드 컴퓨팅

7. 클라우드 컴퓨팅

- **클라우드 컴퓨팅 정의 [NIST SP-800-145]**
  - 네트워크, 서버, 저장 장치, 응용프로그램, 서비스 같은 공유된 컴퓨팅 자원의 모음에 대한 항시적이고 편리하며 필요할 때마다 접근 가능한 모델
  - 서비스의 제공과 종료가 최소한의 관리적인 노력이나 서비스 제공 업체와의 상호작용이 요구
  - 클라우드 모델은 서비스 제공 시간을 늘려주며, 다섯 가지 핵심 특성, 세 가지 서비스 모델, 네 가지 설치 모델이 존재

※ NIST(National Institute of Standards and Technology)
- 미국상무부 기술관리국 산하의 각종 표준과 관련된 기술을 담당하는 연구소

클라우드 컴퓨팅 요소


핵심 특징
광대역 네트워크 접근
빠른 탄력성
측정된 서비스
주문형 셀프 서비스
자원 풀링
서비스 모델
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
배치 모델
공개
개인
혼합
커뮤니티

클라우드 컴퓨팅의 세 가지 서비스 모델

1. **Software as a service (SaaS)**
   - 사용자는 클라우드에서 실행되고 있는 특정 응용프로그램 소프트웨어를 제공 받음
   - 고객이 클라우드 제공자의 기반 시설에서 실행되는 응용프로그램을 사용할 수 있게 함
   - 소프트웨어의 설치, 관리, 업그레이드 및 수정 등의 복잡한 작업을 피하게 해줌

   예) Gmail, Salesforce.com 등

2. **Platform as a service (PaaS)**
   - 사용자는 사용자 응용프로그램이 실행될 수 있는 플랫폼을 제공받음
   - 클라우드 기반 시설에 사용자가 생성하거나 구입한 응용프로그램을 배치하도록 함
   - 유용한 소프트웨어 빌딩 블록과 프로그래밍 언어, 실행 환경, 응용프로그램 설치 도구 같은 여러 개발 도구를 제공
   - 예) Google App Engine, Salesforce1 플랫폼

3. Infrastructure as a service (Iaas)

- 사용자에게 클라우드 기반 시설에 대한 접근을 제공
- API를 통해 제어되는 가상 기계, 추상화된 하드웨어와 운영체제를 제공
- 사용자가 임의의 운영체제와 응용프로그램을 배치하고 실행할 수 있는 처리기, 저장 공간, 네트워크, 그 외 기본적인 컴퓨팅 자원을 제공
- 사용자가 데이터 저장소 같은 기초 컴퓨팅 서비스로 고도의 컴퓨터 시스템을 조합할 수 있게 함

예) Amazon EC2, Windows Azure 등

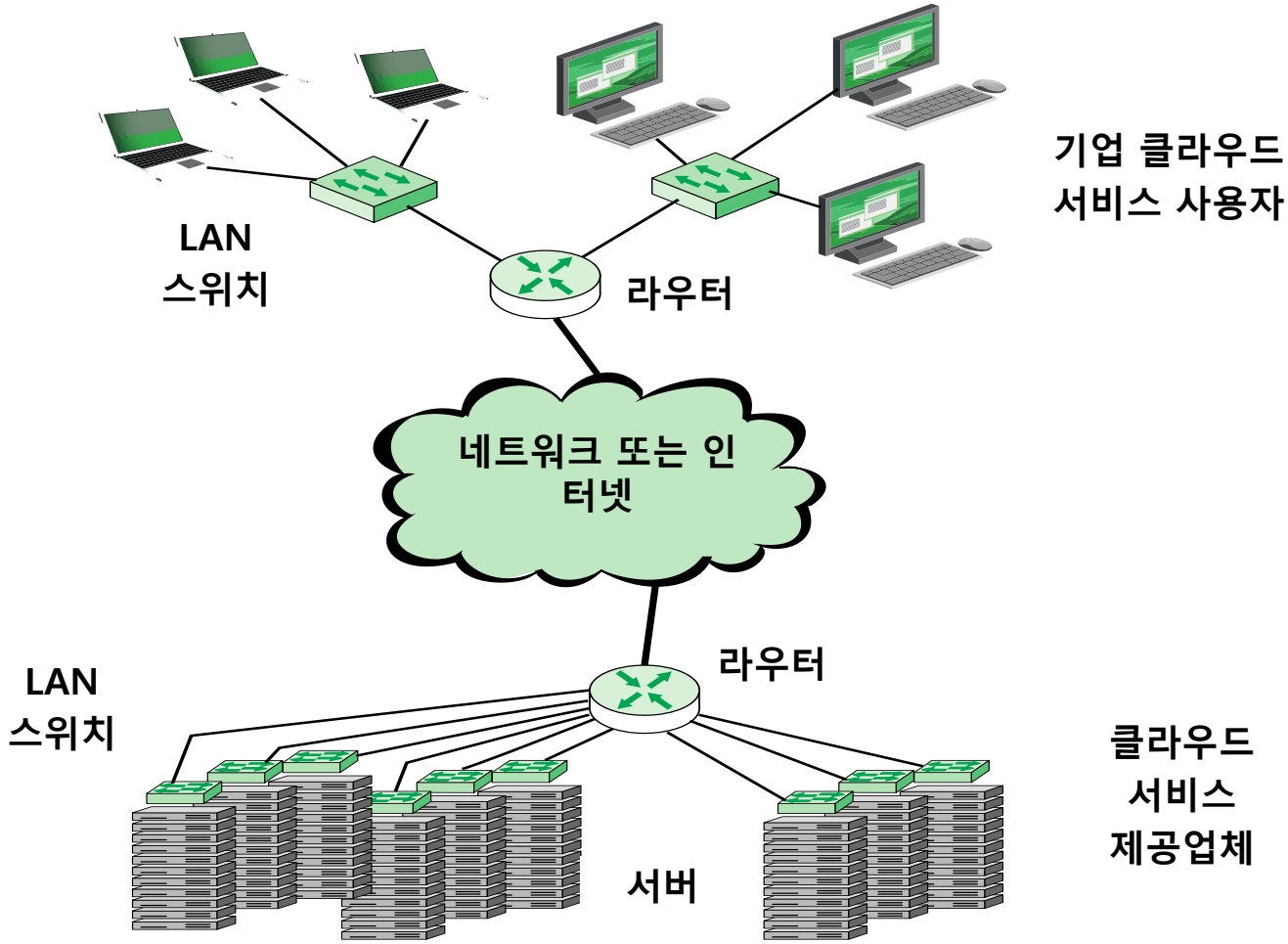
클라우드 컴퓨팅의 배치 모델

1. 공용 클라우드 (Public cloud)
   - 클라우드 시스템이 일반 사용자나 대규모 산업 집단에 제공되고 서비스를 제공하는 한 업체에 의해 소유됨
   - 클라우드 기반 시설과 제어기능 모두 서비스 제공업체의 책임임

2. 사적 클라우드(Private cloud)
   - 클라우드 시스템이 전적으로 하나의 조직에만 제공됨
   - 그 조직이 스스로 관리를 하거나 혹은 제3자가 관리할 수 있고, 기반 시설이 조직 사이트 내에 존재하거나 혹은 외부에 존재할 수 있음
   - 제공 업체는 기반 시설만 책임이 있고 제어는 책임이 없음

3. 커뮤니티 클라우드(Community cloud)
   - 클라우드 시스템이 여러 조직에 의해 공유되고, 미션, 보안 요구 사항, 정책, 적합성 고려사항에 대한 같은 고려사항을 공유하는 특정 집단을 지원하는 경우
   - 조직들과 제3자에 의해 관리될 수 있고 그 위치도 조직 내외부 모두 가능

4. 혼합 클라우드(Hybrid cloud)
   - 다수의 공용, 사적, 혹은 커뮤니티 클라우드 시스템들로 이루어진 클라우드 시스템
   - 각 클라우드는 고유의 시스템으로 남아있으나 데이터와 응용프로그램의 이동성을 지원하는 표준 혹은 비표준기술을 이용해서 서로 긴밀히 연결


기업 클라우드
서비스 사용자
LAN
스위치
라우터
네트워크 또는 인
터넷
라우터
LAN
스위치
클라우드
서비스
제공업체
서버

클라우드 컴퓨팅 관계

클라우드 컴퓨팅 참조 구조

- **[NIST SP 500-292]**
  - "어떻게" 솔루션을 설계하고 구현하는가 대신 클라우드 서비스가 제공하는 "무엇"의 요구 사항에 중점을 둠
  - 참조 구조는 클라우드 컴퓨팅의 동작 복잡성의 이해를 보조

- **NIST 클라우드 컴퓨팅 참조 구조의 목표**
  - 다양한 클라우드 서비스를 전체 클라우드 컴퓨팅 개념적 모델에서 이해하기 쉽도록 보조
  - 고객이 클라우드 서비스를 이해, 논의, 분류, 비교하는 데 기술적 참조를 제공
  - 보안, 상호 운용성, 이식성에 대한 표준과 구현에 대한 분석을 제공

NIST 참조 구조의 행위자

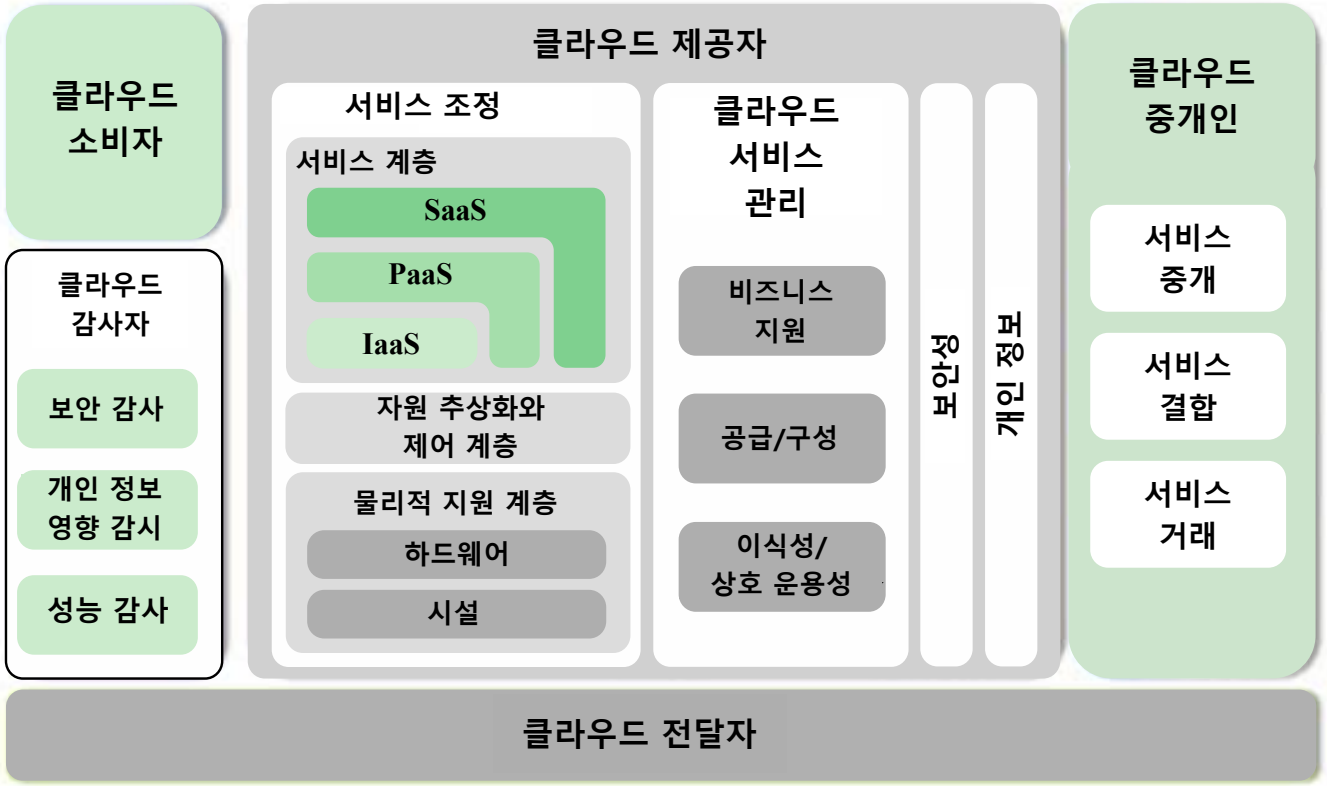
1. 클라우드 소비자(Cloud Consumer)
   - 클라우드 제공자와 비즈니스 관계를 유지하거나 클라우드 서비스를 사용하는 개인이나 조직

2. 클라우드 제공자(Cloud Provider)
   - 이해관계자들을 위해 서비스를 생성하는 책임을 가지는 개인이나 조직

3. 클라우드 감사자(Cloud Auditor)
   - 클라우드 서비스, 정보 시스템 운용, 성능, 클라우드 구현의 보안에 대한 독립적인 평가를 하는 관계자

4. 클라우드 중개인(Cloud Broker)
   - 클라우드 서비스의 사용, 성능, 전달, 클라우드 제공자와 소비자 간의 협상을 관리하는 존재

5. 클라우드 전달자(Cloud Carrier)
   - 클라우드 제공자와 소비자 사이에서 서비스의 연결과 전달을 제공하는 중간자



NIST 클라우드 컴퓨팅 참조 구조

4-8. 클라우드 보안 위험과 대응

8. 클라우드 보안 위험과 대응

- 클라우드 컴퓨팅의 남용과 오용
  - 많은 클라우드 서비스 제공자들의 경우 사용자가 서비스를 등록하고 사용을 시작하기가 비교적 용이하며 어떤 제공자들은 무료 체험 기간 제공함
  - 공격자들은 쉽게 클라우드 시스템 내부에 들어와서 스팸 메일을 보내거나 악성코드 공격을 하거나 DoS 공격 등을 시도해볼 수 있음
  - 대응 방법
    1. 보다 엄격한 초기 등록과 유효성 프로세스
    2. 향상된 신용카드 부정 사용 감시
    3. 고객 네트워크 트래픽에 대한 포괄적인 조사
    4. 네트워크 블록의 공개 블랙리스트 감시

- 비보안 인터페이스와 APIs
  - 제공자는 사용자가 클라우드 서비스를 관리하고 이용하기 위해서 사용하는 소프트웨어 인터페이스 - API를 공개
  - 대응 방법
    1. 제공자 인터페이스의 보안 모델 분석
    2. 강한 인증과 접근 제어, 암호화 전송 구현
    3. API 관련 의존 관계 이해

- 악의적 내부자
  - 내부자의 공격은 통상적인 보안 기법들로 대응하기 어려움
  - 대응 방법
    1. 엄격한 공급망 관리 강화와 포괄적인 공급자 평가 수행
    2. 법적 계약의 일부분으로 인적 자원 요구 명시
    3. 전체 정보 보안과 관리에 투명성과 보고 요구
    4. 보안 침해 알림 프로세스 결정

- 공유 기술 이슈
  - IaaS 업체들은 기반 시설을 공유함으로써 보다 스케일이 큰 서비스를 제공할 수 있으나 공격자가 쉽게 접근 가능하며 보안 관리의 약점
  - 대응 방법
    1. 설치/구성에 대한 최선 보안 습관 구현
    2. 비인가된 수정/행위에 대한 환경 감시
    3. 강한 인증과 관리를 위한 접근과 동작에 대한 접근 제어 증진
    4. SLAs의 취약성 보완에 대한 패치 강화

- 데이터 손실과 누출
  - 가장 치명적인 보안 위험
  - 대응 방법
    1. 강한 API 접근 제어 구현
    2. 전송되는 데이터의 암호화와 무결성 보호
    3. 데이터 보호를 설계와 실행 시 분석
    4. 강한 키 생성, 저장, 관리, 폐기 구현

- 계좌 혹은 서비스 탈취
  - 비밀번호 유출과 함께 가장 위협적인 공격
  - 비밀번호 유출로 인해 공격자는 클라우드 컴퓨팅 서비스의 중요한 부분에 접근
    → 서비스의 비밀성, 무결성, 그리고 접근성을 깨뜨릴 수 있임
  - 대응 방법
    1. 사용자와 서비스 간의 계좌 보증서 공유 금지
    2. 가능한 강한 2 factor 인증 기법 장착
    3. 비인가된 행위 탐지 이후 사후 감시 사용
    4. 클라우드 제공자 보안 정책과 SLAs 이해

- 알려지지 않은 위험 프로파일
  - 클라우드 기반 시설을 사용함에 있어 클라이언트는 보안에 영향이 있을 만한 수많은 이슈에 대해서 클라우드 제공 업체에 일임 하여야 함
  - 클라이언트는 위험 요소를 관리하기 위해서 필요한 역할과 책임을 명료하게 정의하고 주의해야만 함
  - 대응 방법
    1. 가능한 로그와 데이터 공개
    2. 기반 시설(예, 패치 레벨, 방화벽)의 부분/ 전체 공개
    3. 필요한 정보 감시와 경계

4-9. 클라우드 데이터 보호

9. 클라우드 데이터 보호

- 클라우드 데이터에 대한 보안 필요성
  - 많은 위험 요소와 그 요소들 간의 상호작용
  - 클라우드 환경의 구조 및 운영적인 특성

- Multi-instance 모델
  - 각 클라우드 사용자에게 할당된 가상머신에서 독립적으로 실행되는 DBMS를 제공
  - 보안에 관한 역할 정의, 사용자 인증 및 다른 관리 작업에 대해서 완전한 제어 권한을 사용자에게 부여

- Multi-tenant 모델
  - 사용자들에 똑같은 환경을 공유하고 구분을 위해서 데이터를 사용자 ID로 tagging하는 방식
  - Tagging은 데이터베이스를 독점적으로 사용하는 듯한 느낌을 주지만 안전한 데이터베이스 사용 환경을 만들고 유지하는 것은 전적으로 클라우드 제공 업체에 의존하게 됨

- 클라우드 보안에서 SaaS (Software as a Service) 서비스의 일부분으로 SecaaS가 있음

- SecaaS - 클라우드 혹은 클라우드 기반 기반 시설을 통해 보안 응용 프로그램과 서비스를 고객의 시스템에 제공 [Cloud Security Alliance]

클라우드 보안 가이드라인 [NIST SP-800-14]

거버넌스(Governance)	조직의 습관과 관련된 것을 서비스의 설계, 구현, 테스트, 사용, 감시뿐만 아니라 클라우드의 응용프로그램 개발과 서비스 지원의 정책, 절차, 표준으로 확장
컴플라이언스(Compliance)	조직에 보안과 개인 정보 보호 책임을 부과하는 다양한 종류의 법과 규제, 데이터 위치, 개인 정보와 보안 제어, 레코드 관리, 전기적 복구 요구 사항 등을 포함하는 클라우드 컴퓨팅의 잠재적 영향 이해
트러스트(Trust)	서비스 구성이 클라우드 제공자에 의해 배치된 보안과 개인 정보 제어와 처리, 성능에 충분한 가시성을 가지고 있도록 보장 데이터에 대한 명확하고 배타적인 소유권을 확립 시스템의 수명 주기 동안 끊임없이 진화하고 변화하는 위험 환경에서 적응할 만큼 유연한 위험 관리 프로그램을 제정 정보 시스템의 보안 상태의 지속적인 감시로 계속적인 위험 관리 결정을 지원

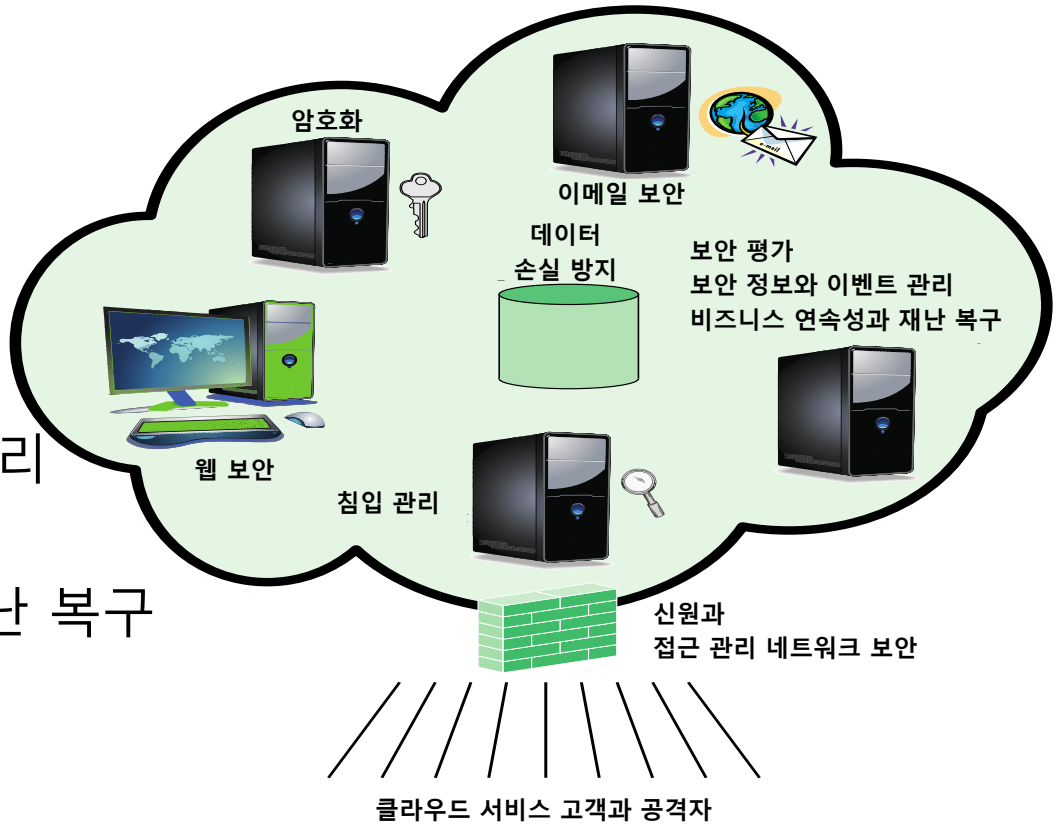
구조(Architecture)	클라우드 제공자가 전체 시스템 컴포넌트들과 시스템 수명 주기 동안 시스템의 보안과 개인 정보에 기술적 제어가 가진 함축적 의미를 포함한 서비스를 제공하기 위해 사용하는 기반 기술들을 이해
신원과 접근 관리(Identity and access management)	적절한 보호 수단이 인증, 공인, 다른 식별과 접근 관리 기능을 위해 있고, 조직의 적합함을 보장
소프트웨어 분리 (Software isolation)	클라우드 제공자가 multitenant 소프트웨어 구조에서 사용하는 가상화와 다른 논리적 분리 기술들을 이해하고, 조직의 위험을 평가
데이터 보호(Data protection)	데이터를 휴식기, 전송, 사용 중 안전 보장을 위해 조직의에 대한 클라우드 제공자의 데이터 관리 솔루션의 적합성, 데이터의 접근 제어 능력을 평가 조직의 데이터를 위협 프로필이 높거나 데이터가 중요한 농축된 값을 나타내는 다른 조직과의 조합에 대한 위험을 고려 클라우드 환경에서 가용한 장비로 암호화키 관리와 제공자에 의한 처리의 위험을 충분히 숙지

가용성 (Availability)	가용성, 데이터 백업과 복구, 재난 복구에 대한 계약 조항과 절차 이해와 이들이 조직의 연속성과 비상계획 요구 사항과 부합함을 보장 붕괴 혹은 심각한 재난 도중에 중요한 조작은 즉시 재개 가능하고, 모든 조작은 결국 시간 내에 조직적으로 재설치됨을 보장
사고 대응 (Incident response)	사고 대응에 관한 계약 조항과 절차를 이해하고 조직의 요구 사항에 부합을 보장 클라우드 제공자가 투명한 대응 프로세스와 사고 중이나 후에 충분한 정보 공유를 하는 메커니즘을 갖도록 보장 조직이 컴퓨팅 환경의 클라우드 제공자의 역할과 책임에 맞추어 조정된 방식으로 사고에 대응하도록 보장

4-10. 서비스로서 클라우드 보안

10. 서비스로서 클라우드 보안

- SecaaS 서비스 카테고리 [Cloud Security Alliance]
  - 신원과 접근 관리
  - 데이터 손실 방지
  - 웹 보안
  - 이메일 보안
  - 보안 평가
  - 침입 관리
  - 보안 정보와 이벤트 관리
  - 암호화
  - 비즈니스 연속성과 재난 복구
  - 네트워크 보안

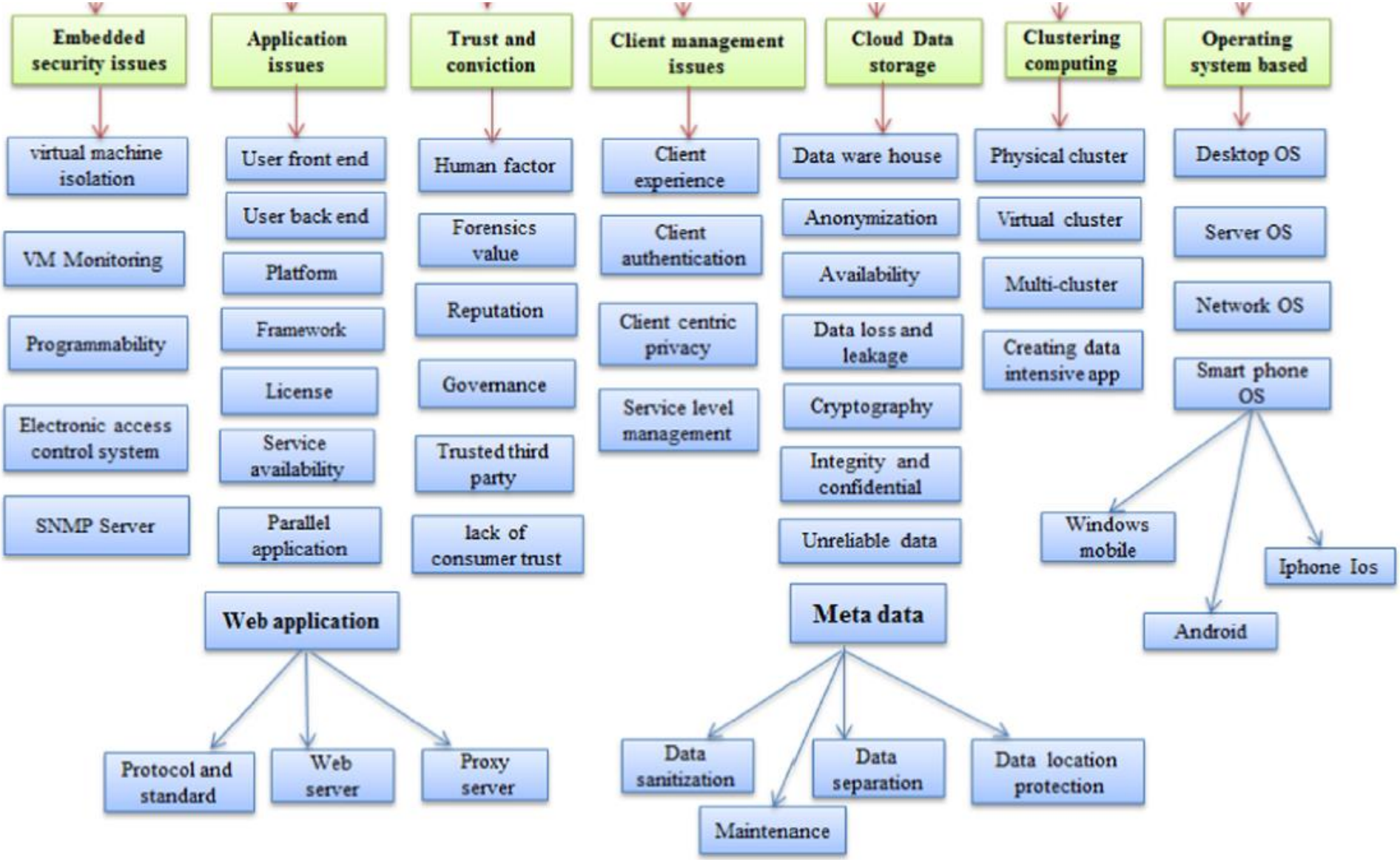


서비스로의 클라우드 보안 요소

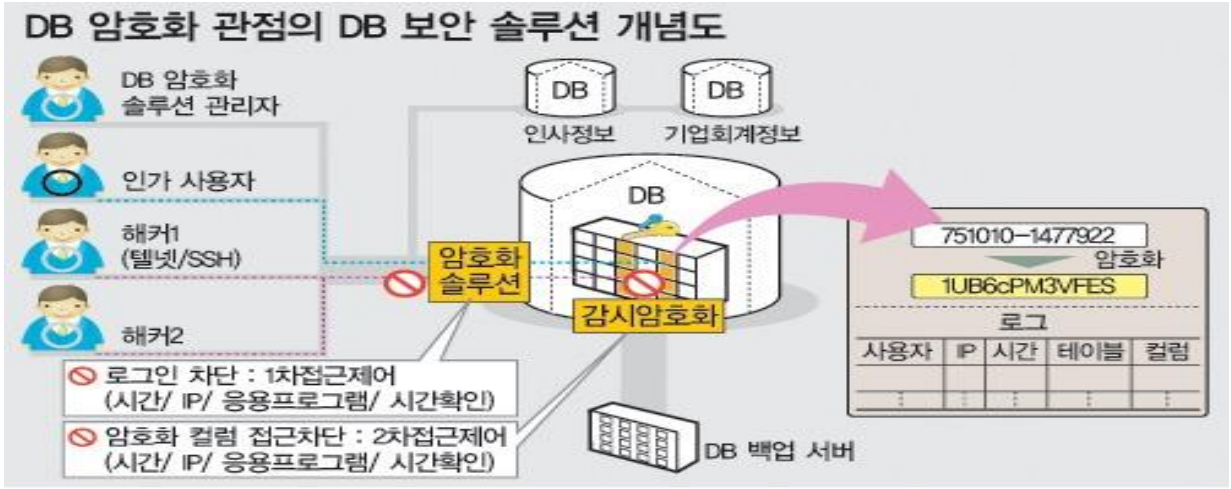
기타 보안 이슈

기타 보안 이슈

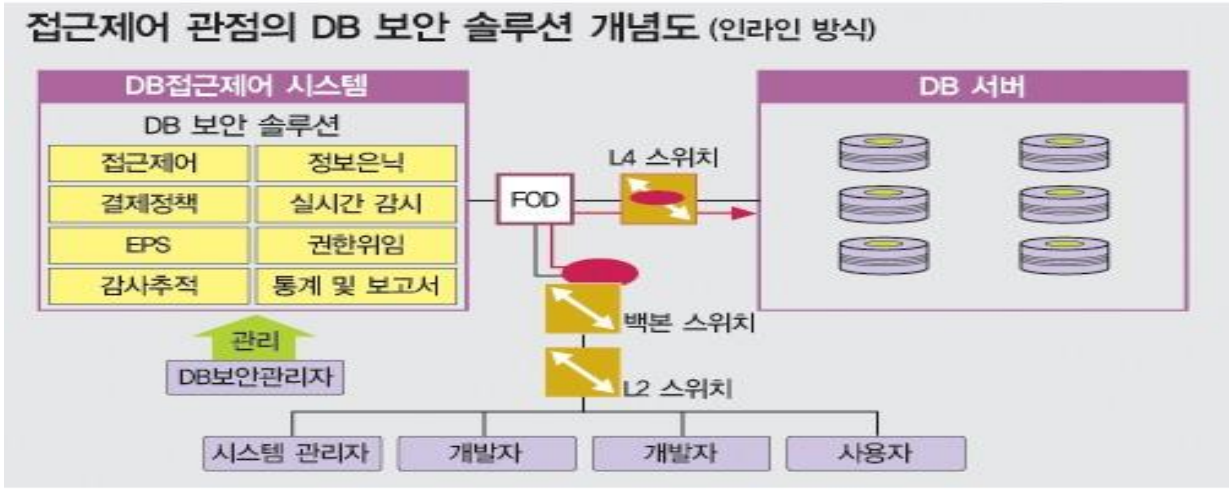
Classification of cloud security issue



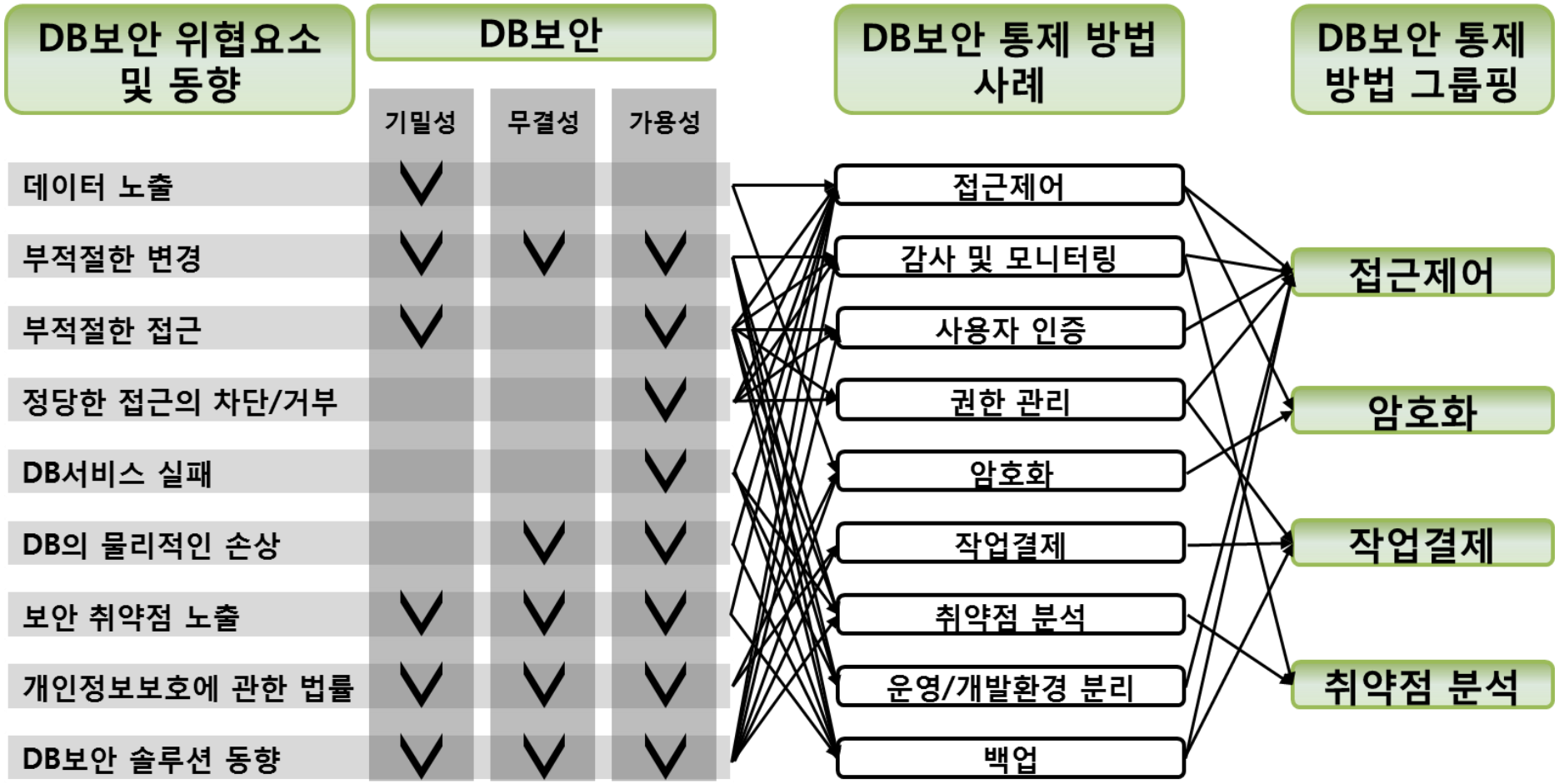
DB 보안 솔루션의 두가지 방식



DB보안 솔루션의 두가지 방식



DB 보안 기술 요소의 도출



참고문헌

- William Stallings Lawrie Brown 저자, 컴퓨터 보안 원리 및 실습 (5판), PEARSON, 2016
- Saurabh Singh, Young-Sik Jeong, Jong Hyuk Park, "A survey on cloud computing security: Issues, threats, and solutions", Journal of Network and Computer Applications, Volume 75, November 2016, Pages 200–222
- http://www.dt.co.kr/contents.html?article_no=2008101002011860713002
- http://www.dbguide.net/db.db?cmd=view&boardUid=152794&boardConfigUid=9&boardIdx=143&boardStep=1

Q & A

부록

관계형 데이터베이스

- 기본 블록은 스프레드시트와 같이 행과 열로 이루어진 데이터 테이블
  - 각 열은 유사한 타입의 데이터를 가짐
  - 각 행은 열의 특정 값을 가짐
  - 테이블은 하나 이상의 유일한 값을 갖고 각 엔트리의 식별자 역할을 하는 열을 가짐

Ex)

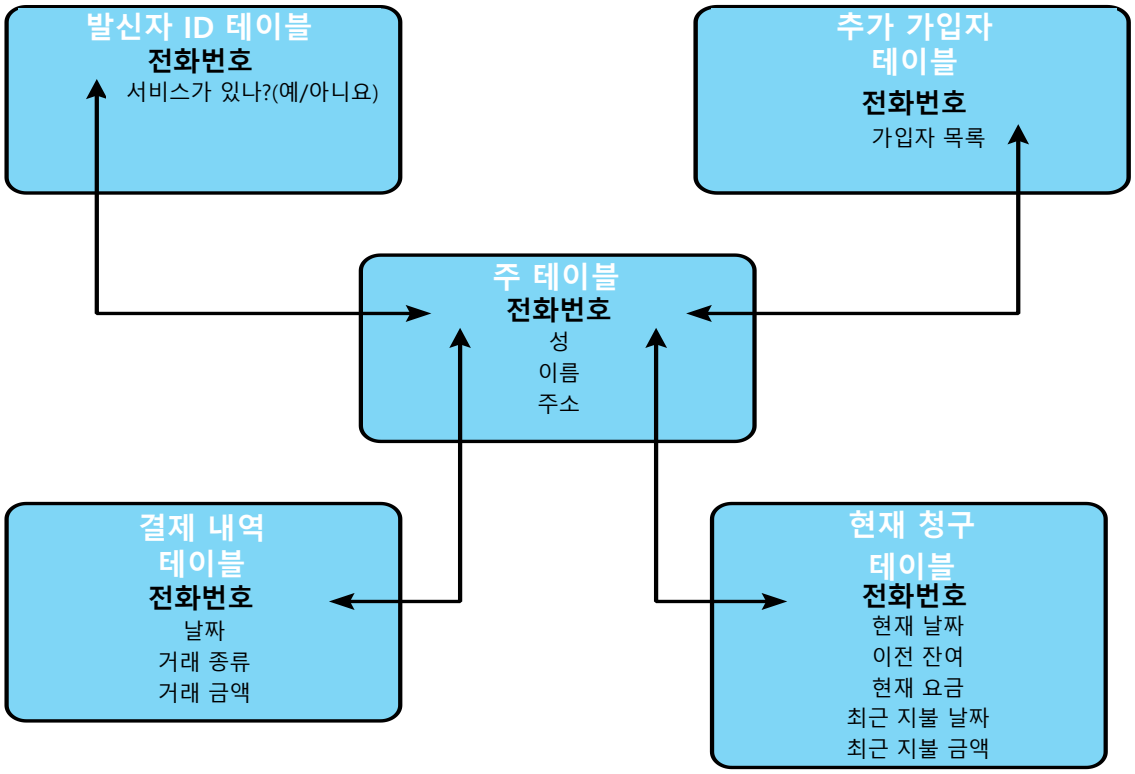
- 전화번호부에서 같은 이름의 가입자들이 다수 존재할 수 있으나 전화번호는 유일하므로 전화번호가 행에 대한 고유한 식별자라 할 수 있음

- 여러 사용자가 하나의 전화번호를 공유할 수 있으므로, 하나의 테이블에 모든 전화번호부의 데이터를 저장하기 위해선 두 번째, 세 번째 가입자를 위한 분리된 열이 필요

- 관계형 데이터베이스 구조는 하나의 고유의 식별자와 묶인 다수의 테이블의 생성을 가능하게 함

- 사용자와 응용프로그램은 데이터베이스 접근을 위해 관계형 쿼리 언어를 사용
  - 쿼리 언어는 프로그램 언어의 절차적 명령보다는 선언적 문장을 사용
  - 쿼리 언어는 사용자들이 주어진 기준에 맞는 데이터의 선택된 아이템들을 모든 레코드에서 추려낼 수 있도록 함

Ex) 전화 회사의 사장은 특정 서비스의 상태, 최근 수납된 요금들에 관한 가입자들의 요금 정보를 추출할 수 있음



관계형 데이터베이스 모델의 예

- 관계형 데이터베이스 시스템의 요소

공식 명칭	공통 명칭	알려진 명칭
관계	테이블	파일
튜플	행	레코드
속성	열	필드

1. 기본키(primary key)
   - 하나 혹은 여러 열로 구성되며, 행의 일부분으로써 테이블 내에서 행을 식별하는 식별자로 쓰임
2. 외래키(foreign key)
   - 두 테이블 사이의 관계를 생성하기 위해서는 한쪽 테이블에서 기본키인 속성이 다른 쪽 테이블에 있어야 하며, 이를 외래키라고 함
3. 뷰(view)
   - 가상 테이블
   - 하나 혹은 여러 테이블에서 쿼리에 의해 선택된 행과 열들임

- 각 속성 A_j는 $|A_j|$ 개 가능한 값을 가지며, x_{ij}는 개인 i의 속성 j의 값을 나타냄

속성

레코드	A_1	• • •	A_j	• • •	A_M
1	x_{11}	• • •	x_{1j}	• • •	x_{1M}
•	•		•		•
•	•		•		•
•	•		•		•
i	x_{i1}	• • •	x_{ij}	• • •	x_{iM}
•	•		•		•
•	•		•		•
•	•		•		•
N	x_{N1}	• • •	x_{Nj}	• • •	x_{NM}

Fi_ 관계형 데이터베이스의 추상 모델

- 구조적 쿼리 언어(SQL)
  - 관계형 데이터베이스에서 스키마 정의, 처리, 데이터 쿼리에 사용
  - 여러 ANSI/ISO 표준과 구현들이 있음

SQL 상태
• 테이블 생성 • 테이블에 데이터 삽입과 삭제 • 뷰 생성 • 쿼리 언어로 데이터 검색

부서 테이블

부서 ID	부서 이름	부서 계정 번호
4	인적 자원	528221
8	교육	202035
9	회계	709257
13	홍보	755827
15	서비스	223945

주 키

직원 테이블

직원 이름	부서 ID	급여 코드	직원 ID	직원 전화번호
Robin	15	23	2345	6127092485
Neil	13	12	5088	6127092246
Jasmine	4	26	7712	6127099348
Cody	15	22	9664	6127093148
Holly	8	23	3054	6127092729
Robin	8	24	2976	6127091945
Smith	9	21	4490	6127099380

외래 키 주 키

(a) 관계형 데이터베이스의 두 테이블

부서 이름	직원 이름	직원 ID	직원 전화번호
인적 자원	Jasmine	7712	6127099348
교육	Holly	3054	6127092729
교육	Robin	2976	6127091945
회계	Smith	4490	6127099380
홍보	Neil	5088	6127092246
서비스	Robin	2345	6127092485
서비스	Cody	9664	6127093148

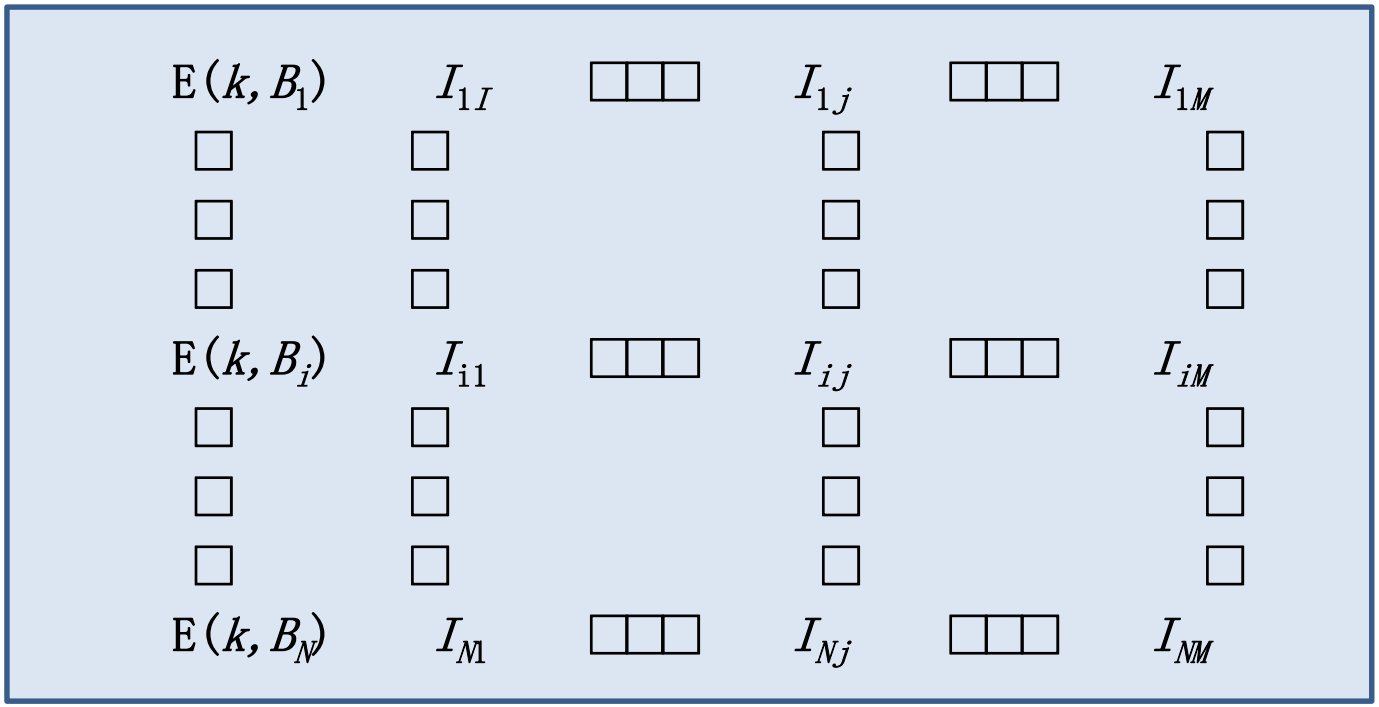
(b) 데이터베이스의 뷰

관계형 데이터베이스 예

데이터베이스 암호화 예

- 데이터베이스 내 테이블의 각 레코드(행)는 블록으로 암호화됨
- 그림 5.3의 관계형 데이터베이스의 추상 모델을 참고하면, 각 행 R_i은 연속적인 블록 $B_i = (x_{i1} \| x_{i2} \| x_{iM})$으로 처리됨
- 텍스트든 숫자든 R_i의 각 속성 값은 비트의 순열로 다루어지고, 행의 모든 속성 값은 하나의 이진 블록을 구성
- 전체 행은 암호화되면 $E(k, B_i) = E(k,(x_{i1} \| x_{i2} \| x_{iM}))$로 표현
- 일부 또는 전체 속성들에 대해 하나의 인덱스 값이 생성
- 암호화되지 않은 데이터베이스의 각 행은 다음과 같이 매핑

$$(x_{i1}, x_{i2}, \ldots, x_{iM}) \longrightarrow [E(k, B_i), I_{i1}, I_{i2}, \ldots, I_{iM}]$$



$B_i = (x_{i1} \,||\, x_{i2} \,||\, x_{iM})$

데이터베이스의 암호화 기법

- employee ID(eid) 값의 범위가 [1,1000]이라 가정
- 값들을 5개 파티션으로 나누고 인덱스 값 1, 2, 3, 4, 5를 할당
  - [1,200], [201, 400], [401,600], [601, 800], [801, 1000]
- 텍스트 필드에 대해 속성 값의 첫 글자로 인덱스를 도출
- 속성 ename에 대해서는, A로 시작하면 인덱스 1, B는 2, …같이 할당
- 각 속성에 이와 유사한 파티션 기법을 사용
- 첫 열의 값은 각 행의 암호화된 값을 나타냄
- 실제 값은 암호화 알고리즘과 암호화키로 결정
- 나머지 열들은 해당되는 속성 값들에 대한 인덱스 값을 보여줌

(a) Employee 테이블

eid	ename	salary	addr	did
23	Tom	70K	Maple	45
860	Mary	60K	Main	83
320	John	50K	River	50
875	Jerry	55K	Hopewell	92

(b) 암호화된 Employee 테이블과 인덱스

E(k,B)	I(eid)	I(ename)	I(salary)	I(addr)	I(did)
1100110011001011…	1	10	3	7	4
0111000111001010…	5	7	2	7	8
1100010010001101…	2	5	1	9	5
0011010011111101…	5	5	2	4	9

암호화된 데이터베이스 예

- 속성 값과 인덱스 값 사이의 매핑함수는 고객에 저장된 메타데이터, 데이터 소유자 위치로 구성됨

- 이러한 배치는 보다 효과적인 데이터 검색을 제공

  Ex) 사용자가 eid < 300인 모든 고용인 레코드를 요청한다고 가정하면, 쿼리 처리기는 I(eid) <= 2인 모든 레코드를 요청함. 이는 서버에 의해 반환됨. 쿼리 처리기는 반환된 모든 행을 복호화하고, 원래 쿼리와 맞지 않는 것은 폐기하고 요청된 비암호화된 데이터를 사용자에 반환

- 기술된 인덱싱 기법은 공격자에게 어느 정도의, 주어진 속성에 대한 대략적인 상대적 행의 순서같은, 정보를 제공함

- 정보를 감추기 위해서, 인덱스의 순서는 임의로 배치될 수 있음

  Ex) eid는 [1,200], [201, 400], [401,600], [601, 800], [801, 1000]로 파티션이 되어 2, 3, 5, 1, 4로 매핑될 수 있음. 서버에는 메타데이터가 저장되지 않기 때문에, 공격자는 서버로부터 정보를 얻을 수 없음

클라우드 컴퓨팅의 핵심 특성

1. 폭넓은 네트워크 접근
   - 네트워크를 통해서 서비스에 접근 가능하고, 전통적인 혹은 클라우드 기반 소프트웨어 서비스뿐만 아니라, 이질적인 여러 클라이언트 플랫폼(모바일폰, 랩톱, PDA 등)에서 사용을 장려하는 표준화된 기술을 통해 접근 가능

2. 신속한 신축성
   - 클라우드 컴퓨팅은 사용자의 개별 서비스 요구 사항에 맞추어 사용자가 자원을 늘리고 줄일 수 있는 능력을 부여함

3. 조정된 서비스
   - 클라우드 시스템은 저장 장치, 처리기, 통신 자원, 활성 중인 사용자 계정 등 서비스 종류에 적합한 추상 단계에 따른 계측 기능을 활용하여 자원을 자동으로 제어하고 최적화 가능

4. 주문형 셀프 서비스
   - 소비자는 서버 시간이나 네트워크 저장 장치 같은 리소스를 사용자와의 상호작용 없이 자동으로 필요한 만큼 제공받을 수 있음

5. 자원 풀링
   - 제공 업체의 컴퓨팅 자원은 여러 소비자에게 다중-세입자 모델을 이용하여 저장되는데, 다양한 물리적 혹은 가상 자원들이 사용자의 요구에 따라 동적으로 할당과 재할당을 받음

   - 소비자는 보통 제공된 자원의 정확한 위치를 알 수 없고 제어할 수도 없지만 국가, 주, 데이터센터 같은 고차원의 위치를 지정할 수도 있다는 점에서 어느 정도 위치 독립성의 수준이 있다고 할 수 있음