네트워크 보안 에센셜

- 4장 키 분배와 사용자 인증 –

명 세인(sein@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

대칭암호의 키 분배

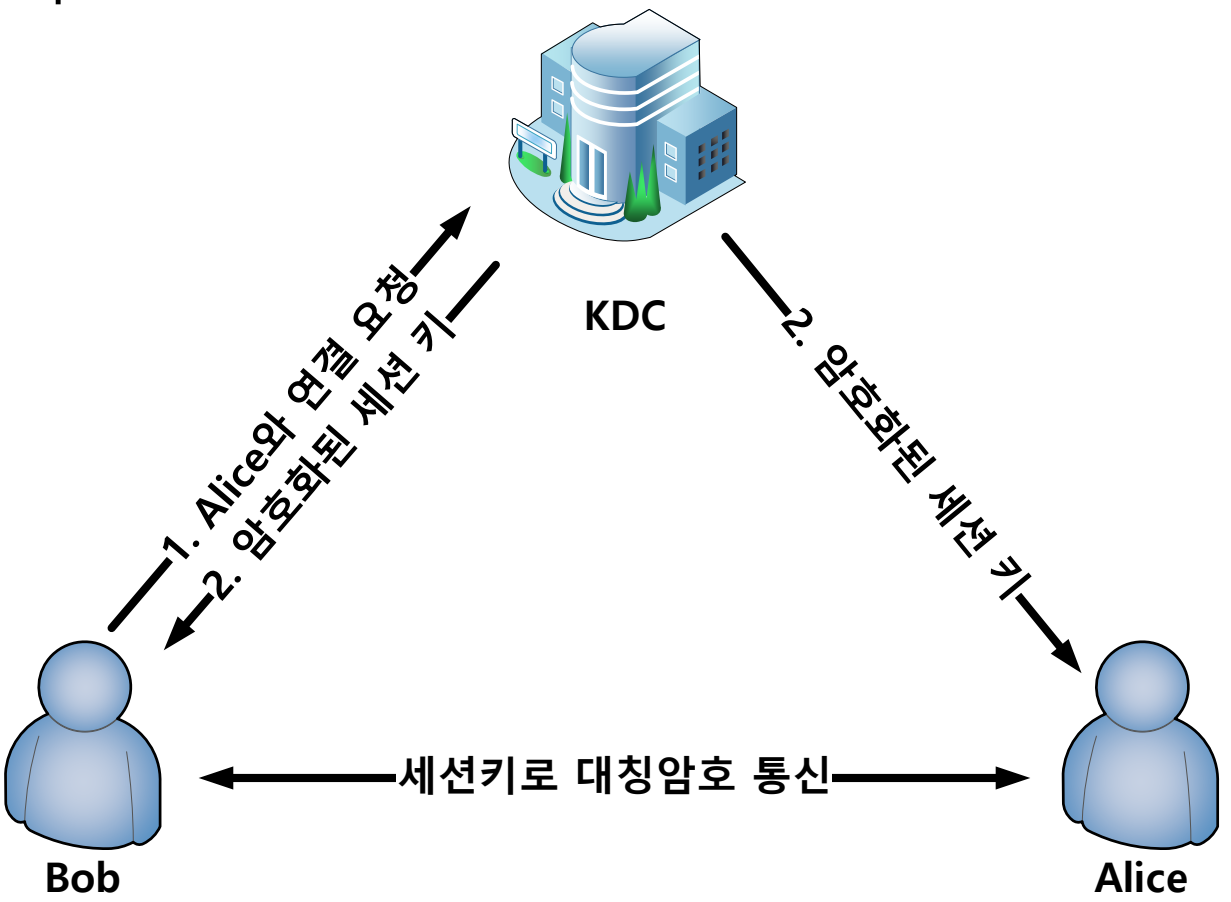
- 대칭 암호는 키를 비밀로 하여 기밀성을 보장하므로 키 분배 시 보안을 유지 하고 그 방법으로 4가지 정도로 분류
- Alice와 Bob이 대칭 키를 갖는 방법
  1. Alice가 키 쌍을 생성하여 Bob에게 직접(물리적) 전달
  2. Darth가 키 쌍을 생성하여 Alice와 Bob에게 직접 전달
  3. Alice와 Bob이 최근에 상용했던 키가 있다면 새 키 쌍을 생성하여 이전 키로 암호화 하여 전송
  4. Darth가 Alice와 Bob 각각에게 암호화 연결이 확립되어 있는 경우, Darth가 키 쌍을 생성하여 각 각 전달

대칭암호의 키 분배

- Alice와 Bob이 대칭 키를 갖는 방법
  - 1번과 2번의 경우
    - 링크 암호화에 적합, 종단간 암호화에 적용하기 어려움
  - 3번의 경우
    - 링크암호, 종단간 암호화에 사용가능 하지만 공격자가 사용된 키 중 한 개라도 얻으면 이후의 키는 모두 노출
  - 4번의 경우
    - 두 가지 키를 사용하는 개념으로 키 분배 센터(KDC: Key Distribution Center)가 통신 권한을 결정하고 영구 키로 일회용 세션 키를 생성
      - Permanent Key: 개체에게 세션 키를 분배하기 위한 키
      - Session Key: 통신하기 위해 구성된 세션에서만 사용하기 위한 일회용 키

대칭암호의 키 분배

- Alice와 Bob이 대칭 키를 갖는 방법
  - 4번의 경우
    - KDC의 동작



목 차

- 대칭 암호를 이용한 대칭키 분배
- KERBEROS
- 비대칭 암호를 이용한 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합 신원 관리

KERBEROS

- 개요
  - MIT에서 Project Athena 를 통해 개발되었고, 버전 1~3은 MIT 내부적으로만 존재
  - KERBEROS는 개방형 분산 환경인 서버-클라이언트 모델에서 상호 인증을 제공 하기 위한 프로토콜
    - 대칭키 암호
    - 서버-클라이언트 상호 인증
      - 재전송 공격 방지
    - KERBEROS는 프로토콜 동작 시 UDP 88을 사용

KERBEROS

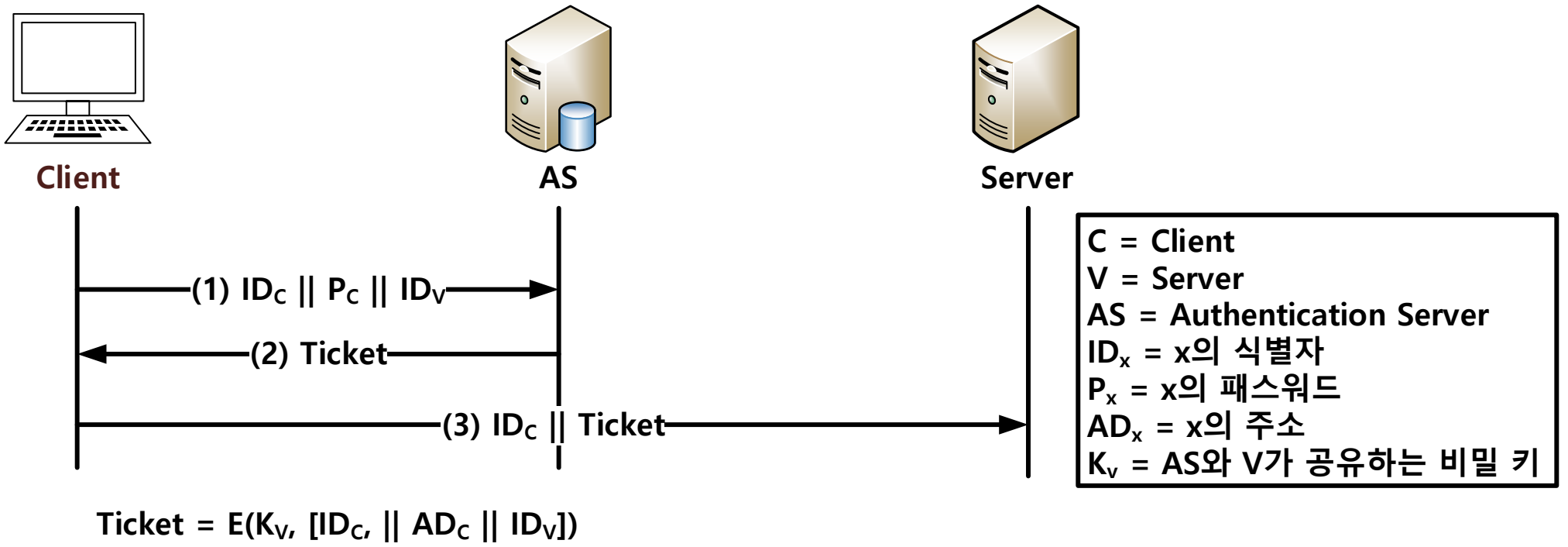
- 개요
  - Single Sign On기반으로 한 세션의 인증을 수립하면 이후 작업을 보안처리
  - 사용자 컴퓨터에서 사용자 인증을 못하는 경우 가능한 공격
    - 사용자 신분 위장
    - 위장 서버
    - 통신 갈취 후 재전송을 통한 접근 권한 획득 또는 가용성 침해

KERBEROS

- 기본적인 인증의 절차
  - 단순 인증 절차
    - 서비스 서버가 인증을 수행하기에는 부담이 됨, 따라서 사용자의 패스워드를 데이터베이스에 저장하는 인증 서버 (AS: Authentication Server)를 이용
  - 보다 안전한 인증 절차
    - 단순 인증에서 기본적인 인증을 해결, 하지만 패스워드의 입력을 최소화하며 패스워드를 평문으로 전송하지 않아야 함
      - 평문형 패스워드를 사용하지 않는 구조와 티켓 발행 서버(TGS: Ticket-Granting Server)개념을 도입
  - KEBEROS는 보다 안전한 인증 절차에 재전송 공격을 방지

KERBEROS

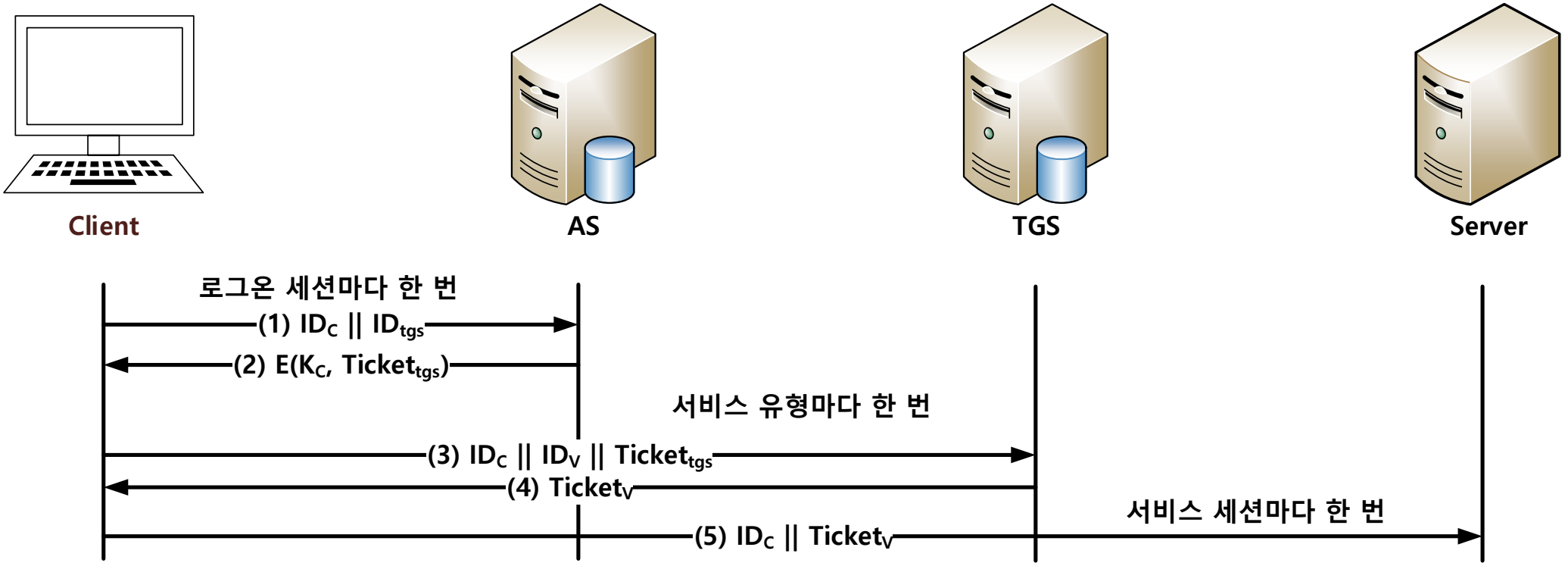
- 기본적인 인증의 절차
  - 단순 인증 절차



  - 서버는 티켓을 복호화 하여 (3)메시지와 비교하여 클라이언트 인증을 수행

KERBEROS

- 기본적인 인증의 절차
  - 보다 안전한 인증 절차



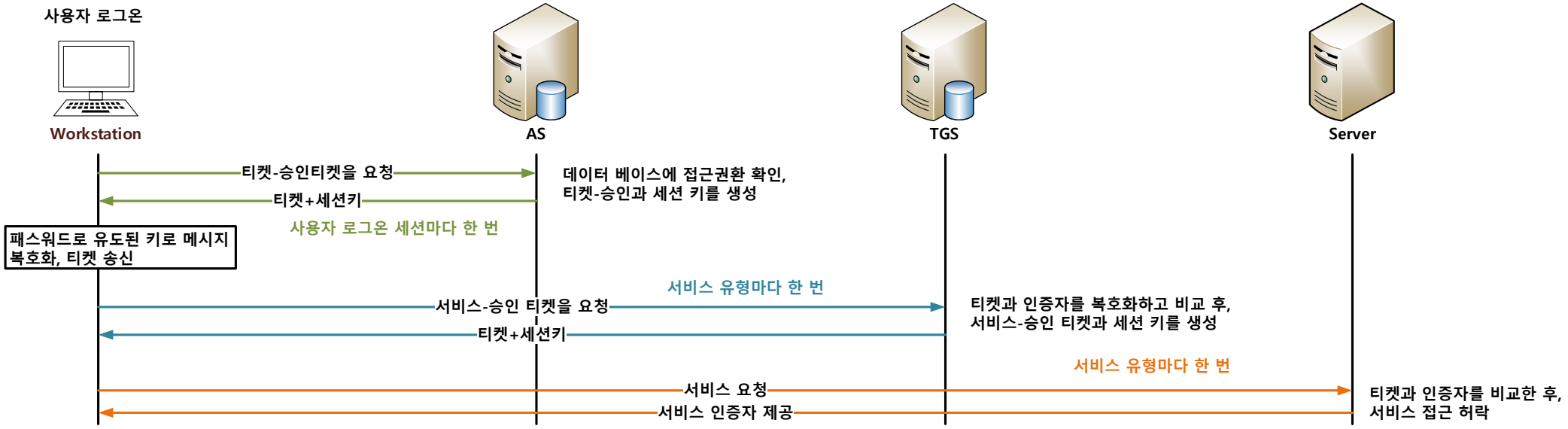
$Ticket_{tgs} = E(K_{tgs}, [ID_C \| AD_C \| ID_{tgs} \| TS_1 \| Lifetime_1])$
$Ticket_V = E(K_V, [ID_C \| AD_C \| ID_V \| TS_2 \| Lifetime_2])$

KERBEROS

- KERBEROS v4
  - 역사
    - Steve Miller와 Clifford Neuman이 1980년대에 발표
  - 인증에 추가된 개념
    - 각 티켓의 유효 기간
    - 서버의 인증
    - 공동체

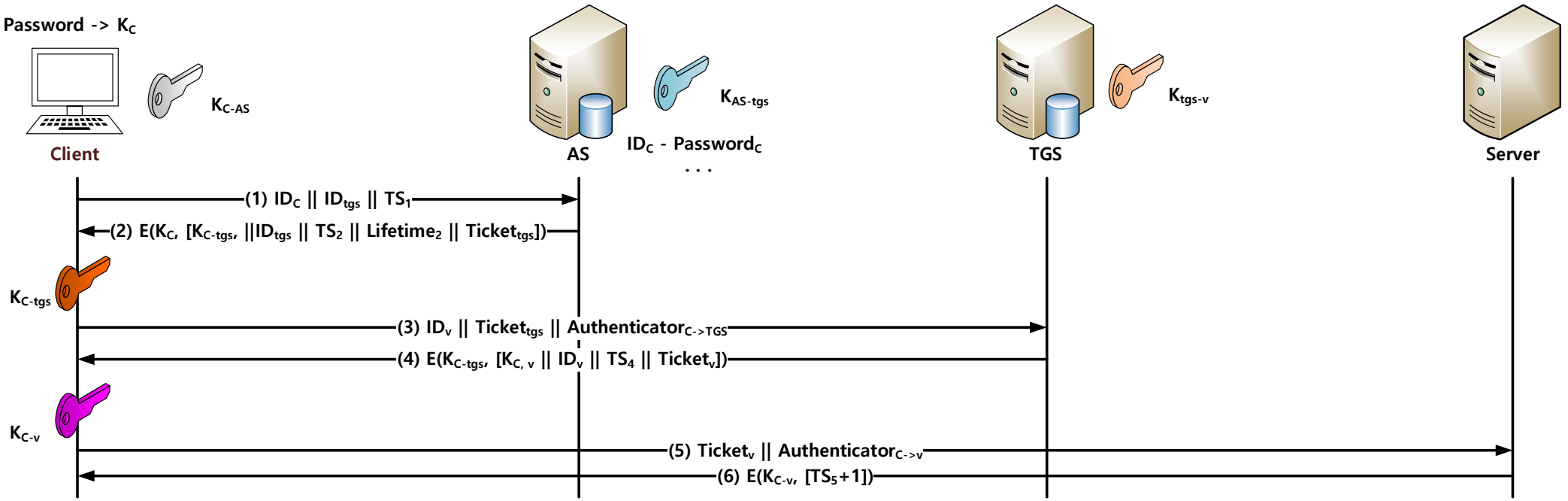
KERBEROS

- KERBEROS v4
  - 인증 절차 개요
    - 서버를 인증하여 위장서버 공격 방지
    - 티켓 사용자가 정당한 티켓 발행자인지 각 서버가 인증



KERBEROS

- KERBEROS v4
  - 인증 절차 메시지



(2,3): $Ticket_{tgs} = E(K_{AS-tgr}, [K_{C-tgs} || ID_C || AD_C || ID_{tgs} || TS_2 || Lifetime_2])$

(3): $Authenticator_{C->TGS} = E(K_{C-tgs}, [ID_C || AD_C || TS_3 || Lifetime_3])$

(4,5): $Ticket_v = E(K_{tgs-v}, [K_{C-v} || ID_C || AD_C || ID_v || TS_4 || Lifetime_4])$

(5): $Authenticator_{C->v} = E(K_{C-v}, [ID_C || AD_C || TS5])$

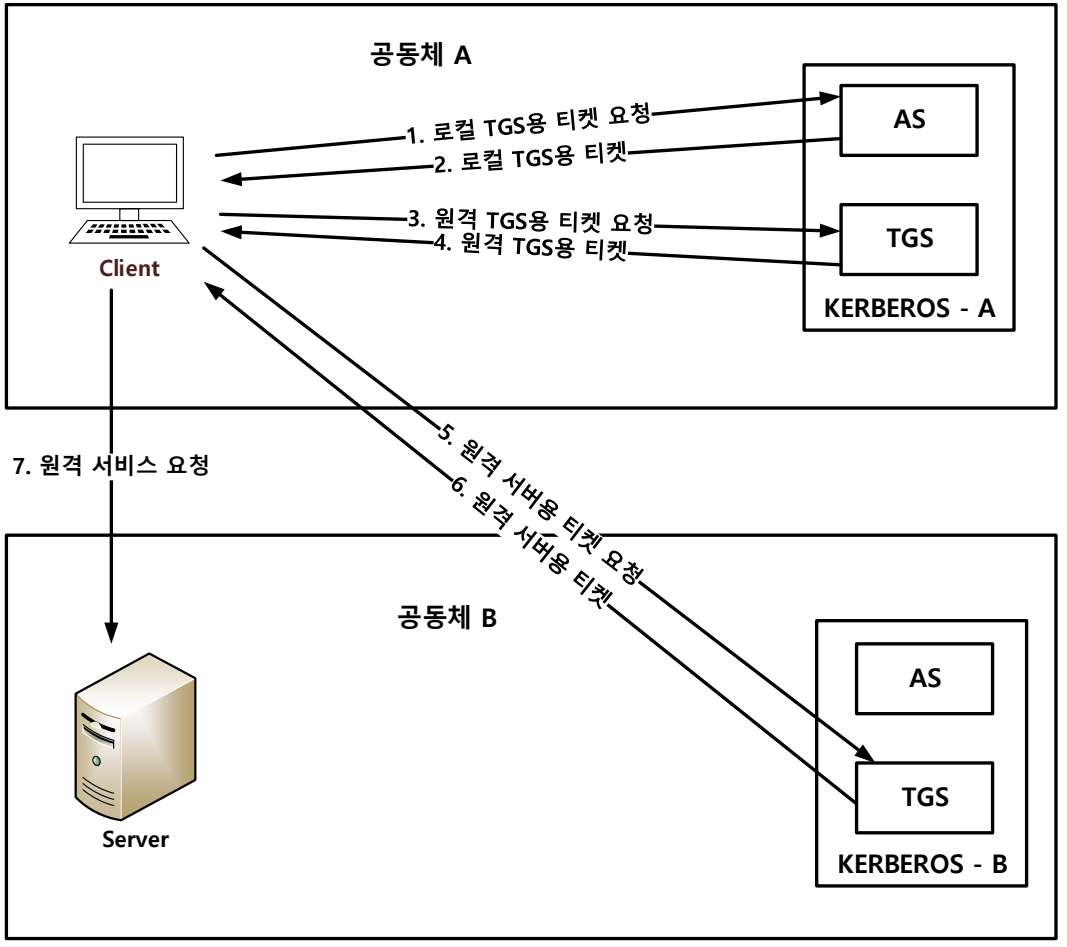
KERBEROS

- KERBEROS v4
  - 다중 KERBEROS
    - KERBEROS 프로토콜의 AS, TGS, 다수의 클라이언트, 다수의 서비스 서버로 구성된 완전한 구현을 KEBEROS Realm(공동체)라 함
      - KERBEROS AS는 반드시 ID와 모든 사용자의 패스워드 해시를 데이터베이스에 저장, 모든 사용자는 KERBEROS AS에 등록
      - KERBEROS AS는 필히 각 서버(TGS, 서비스 서버)와 비밀키를 공유, 모든 서버는 KERBEROS AS에 등록
    - Realm간 인증
      - 교류하는 Realm에 속한 KERBEROS AS는 다른 Realm의 AS와 비밀키를 공유, 두 KERBEROS AS는 상호간 등록

KERBEROS

- KERBEROS v4
  - 다중 KERBEROS
    - 공동체간 인증

1. $ID_C \| ID_{tgs} \| TS_1$
2. $E(K_C, [K_{C\text{-}tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$
3. $ID_{tgsrem} \| Ticket_{tgs} \| Authenticator_C$
4. $E(K_{C\text{-}tgs}, [K_{C\text{-}tgsrem} \| ID_{tgsrem} \| TS_4 \| Ticket_{tgsrem}])$
5. $ID_{Vrem} \| Ticket_{tgsrem} \| Authenticator_C$
6. $E(K_{C\text{-}tgsrem}, [K_{C\text{-}Vrem} \| ID_{Vrem} \| TS_6 \| Ticket_{Vrem}])$
7. $Ticket_{Vrem} \| Authenticator_C$



KERBEROS

- KERBEROS v4
  - 환경적 결함
    - Athena 프로젝트 환경에 국한하여 사용하는 것이 원래 목적
      - Encryption System Dependence: 이 버전에서는 DES를 사용, 수출제한과 보안강도 문제가 있음 v5에서 암호문에 암호유형 식별 추가
      - Internet Protocol Dependence: 이 버전에서는 IP주소만 사용 가능, (ISO네트워크 불가능) v5에서 네트워크 주소의 유형과 길이를 표시할 수 있음
      - Message Byte Ordering: 바이트순서를 결정하고 그것을 표시, v5에서는 ASN.1(Abstract Syntax Notation One)과 BER(Basic Encoding Rules) 사용
      - Ticket Lifetime: 이 버전에서 유효기간은 5분단위로 8비트 사용, 한계가 있음 v5에서는 시작과 종료시간을 명시
      - Authentication Forwarding 다른 호스트로 복사된 같은 인증서를 사용할 수 없음, v5에서 가능
      - Interrealm Authentication: v4에서는 N개의 공동체는 N^2수준의 관계가 필요, v5에서는 적은 수의 관계로 구현

KERBEROS

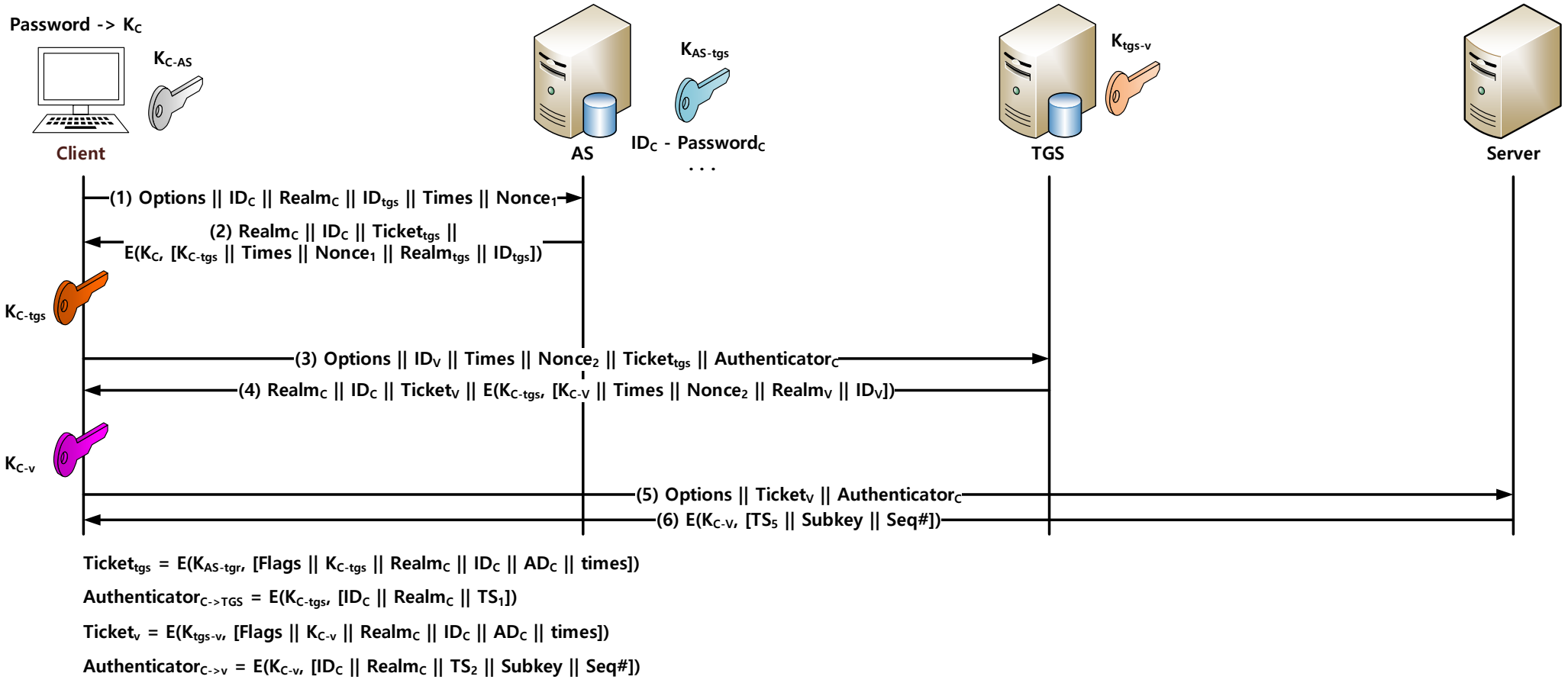
- KERBEROS v4
  - 기술적 결함
    - 자체적인 기술적인 문제
      - Double Encryption: 티켓을 전송할 때 두 번 암호화됨
      - PCBC 암호화: 이 버전에서는 비 표준 DES 암호블록 운용 모드를 사용 취약점이 증명됨, v5는 CBC모드로 무결성 메커니즘 제공
      - Session Keys: 동일한 티켓이 반복적으로 사용될 수 있으며 클라이언트, 서버 대상으로 재전송 공격 가능, v5에서는 세션키 협상 가능
      - Password Attacks: v4, v5모두 이 공격에 취약함, AS에서 Client로 가는 메시지는 패스워드 기반의 키 암호화 정보

KERBEROS

- KERBEROS v5
  - 개요
    - 역사
      - 1993년 John Kohl 과 Clifford Neuman이 RFC 1510에 나타냄
      - KERBEROS v4의 한계와 보안 문제를 보완
    - 추가된 요소
      - Realm: 사용자의 공동체를 의미
      - Options: 돌아오는 티켓의 특정 플래그의 설정을 요청
      - Times: 클라이언트가 티켓 안의 시간 설정을 요청
        - From: 언제부터 사용
        - Till: 만료시간
        - Rtime: 요구되는 재시작부터 만료시간까지
      - Nonce: 두 번째 메시지에서 반복 사용되는 랜덤넘버, 응답이 재전송된 것이 아님을 확신

KERBEROS

- KERBEROS v5
  - 인증 절차



$Ticket_{tgs} = E(K_{AS-tgr}, [Flags || K_{C-tgs} || Realm_C || ID_C || AD_C || times])$

$Authenticator_{C->TGS} = E(K_{C-tgs}, [ID_C || Realm_C || TS_1])$

$Ticket_v = E(K_{tgs-v}, [Flags || K_{C-v} || Realm_C || ID_C || AD_C || times])$

$Authenticator_{C->v} = E(K_{C-v}, [ID_C || Realm_C || TS_2 || Subkey || Seq\#])$

KERBEROS

- KERBEROS v5
  - 인증자에 추가된 요소
    - Subkey: 특정 응용 세션을 보호하기 위해 사용할 암호화 키, 클라이언트가 선택, (Default: K_{C-V})
    - Sequence Number: 선택사항 필드, 시작 순서번호를 명시

KERBEROS

- KERBEROS 프로토콜
  - 장점
    - 사용자와 서버간 통신에 대칭키를 사용한 기밀성과 무결성 보장
  - 단점
    - 키 분배 센터에 오류가 발생하거나 공격 당하면 전체 서비스 다운
    - 사용자의 패스워드 공격에 취약하고 유출 탐지가 어려움
    - 사용자 패스워드 변경시 비밀키도 변경

목 차

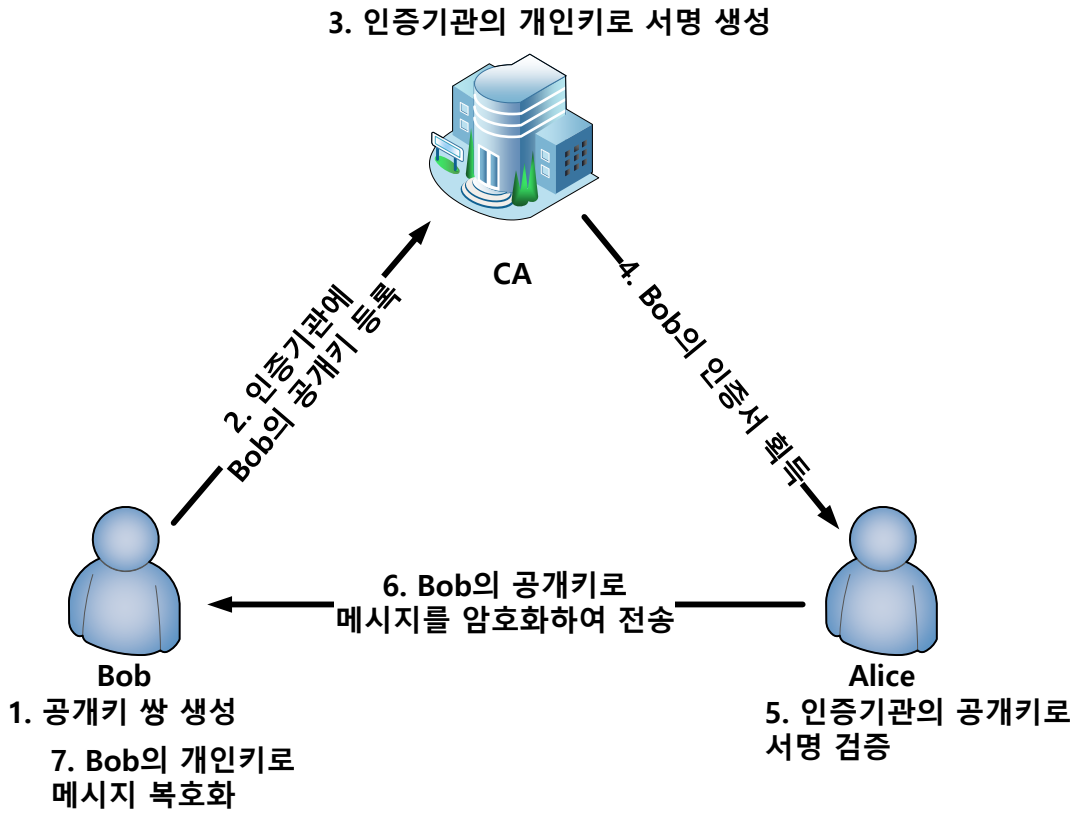
- 대칭 암호를 이용한 대칭키 분배
- Kerberos
- 비대칭 암호를 이용한 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합 신원 관리

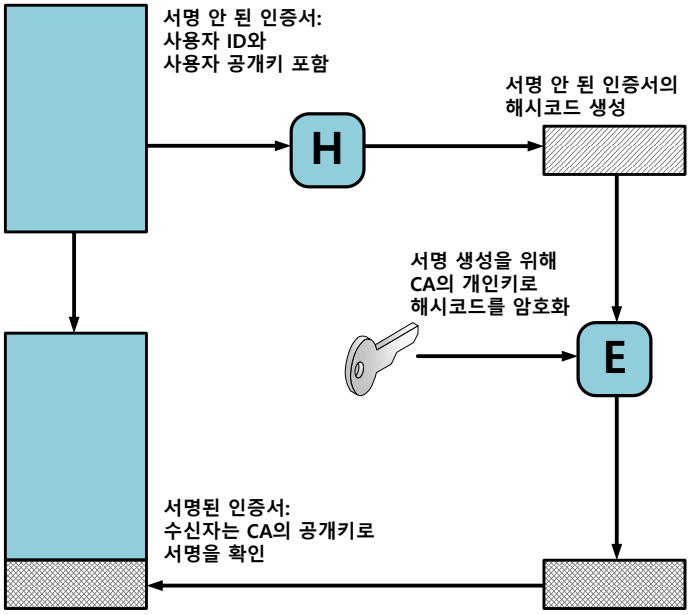
비대칭 암호를 이용한 키 분배

- 공개키 암호의 중요 역할은 암호 키의 분배
  - 공개키 분배
    - 공개키 인증서
      - 공개키를 공개하는 사용자를 신분위장하여 가짜 공개키를 공개함을 방지하기 위한 공개키 암호 사용시의 인증 방법
      - 인증서는 공개키와 키 소유자의 ID로 구성되고 이에 대해 신뢰할 만한 제 3자(인증기관) 디지털 서명을 한 것을 의미
        - 인증서의 사용으로 X.509 인증서 개념이 표준
        - 인증기관 (CA: Certificate Authority)
          - 정부기관 또는 금융기관등

비대칭 암호를 이용한 키 분배

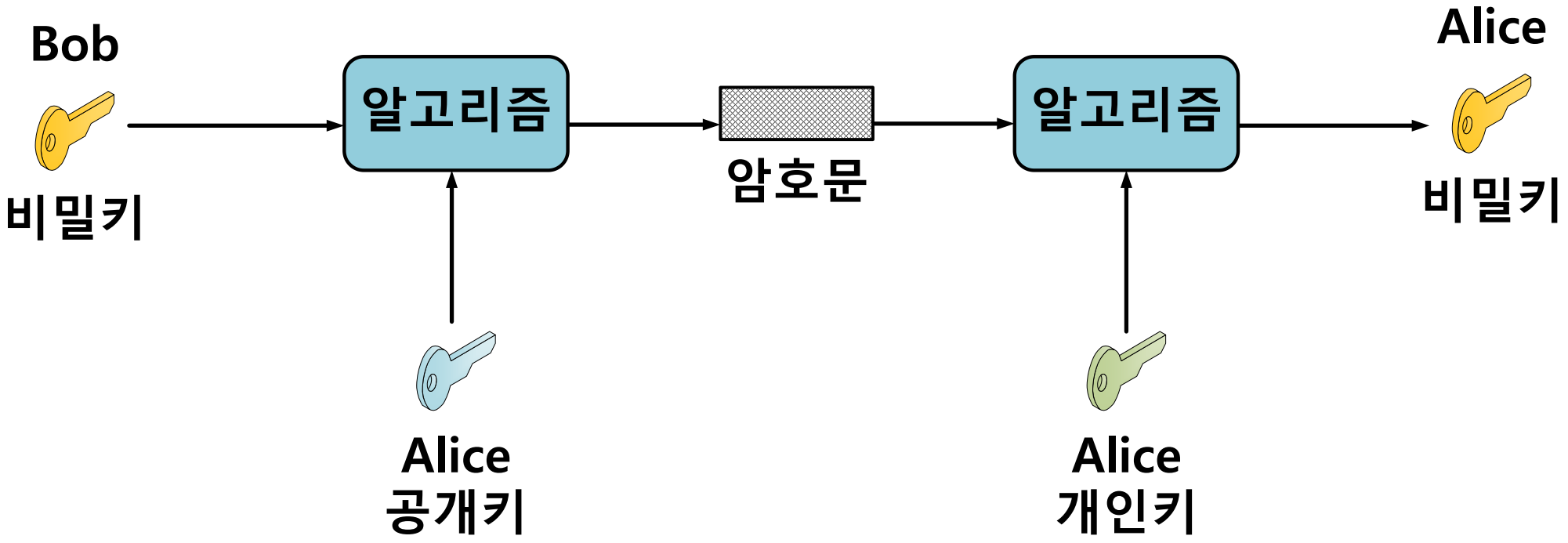
- 공개키 암호의 중요 역할은 암호 키의 분배
  - 공개키 분배
    - 공개키 인증서





비대칭 암호를 이용한 키 분배

- 공개키 암호의 중요 역할은 암호 키의 분배
  - 공개키를 이용한 비밀키 분배
    - 공개키 암호를 사용하여 비밀키를 전송



목 차

- 대칭 암호를 이용한 대칭키 분배
- Kerberos
- 비대칭 암호를 이용한 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합 신원 관리

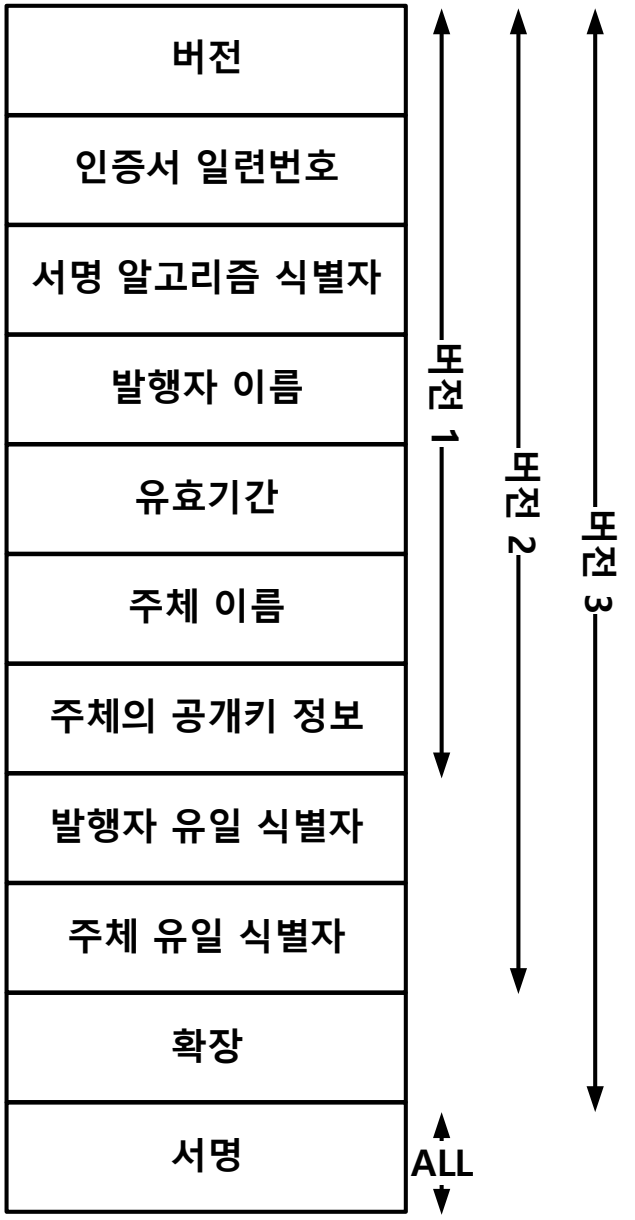
X.509 인증서

- 개요
  - 역사
    - 1988년 X.500 표준안에서 파생
    - 1993년 v2, 1996년 확장기능이 추가된 v3를 발표
    - 공개키 인증서로 매우 중요한 역할을 하는 중
  - 공개키 암호(RSA권고)와 디지털 서명(해시)을 이용

X.509 인증서

- X.509 인증서 형식
  - 기본형식과 X.509 v1 설명 표

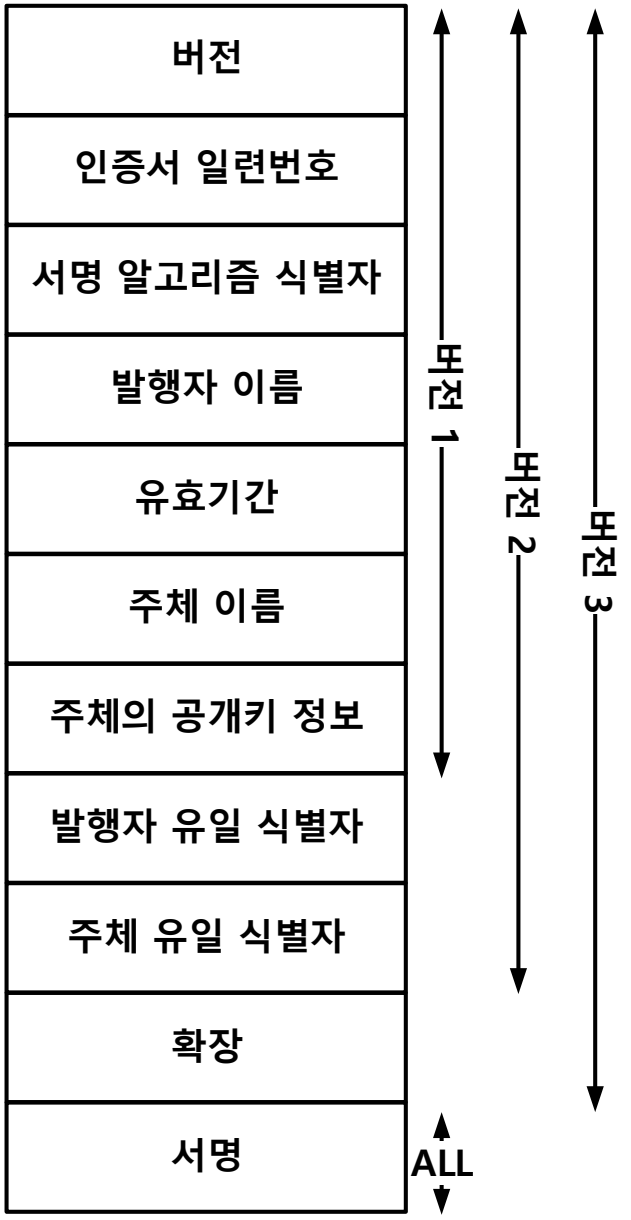
이름	의미
Version	인증서 형식(버전) 구별
Serial Number	인증서 식별 번호
Signature Algorithm ID	인증서 서명에 사용한 알고리즘 식별
Issuer Name	인증서를 발행하고 서명한 CA의 이름
Period of Validity	시작일~종료일
Subject Name	인증서가 인증하는 사용자 이름
Subject's Public-key Info	사용자의 공개키와 적용되는 알고리즘 식별자
Signature	인증기관의 개인키 서명



X.509 인증서

- X.509 인증서 형식
  - X.509 v2 설명 표

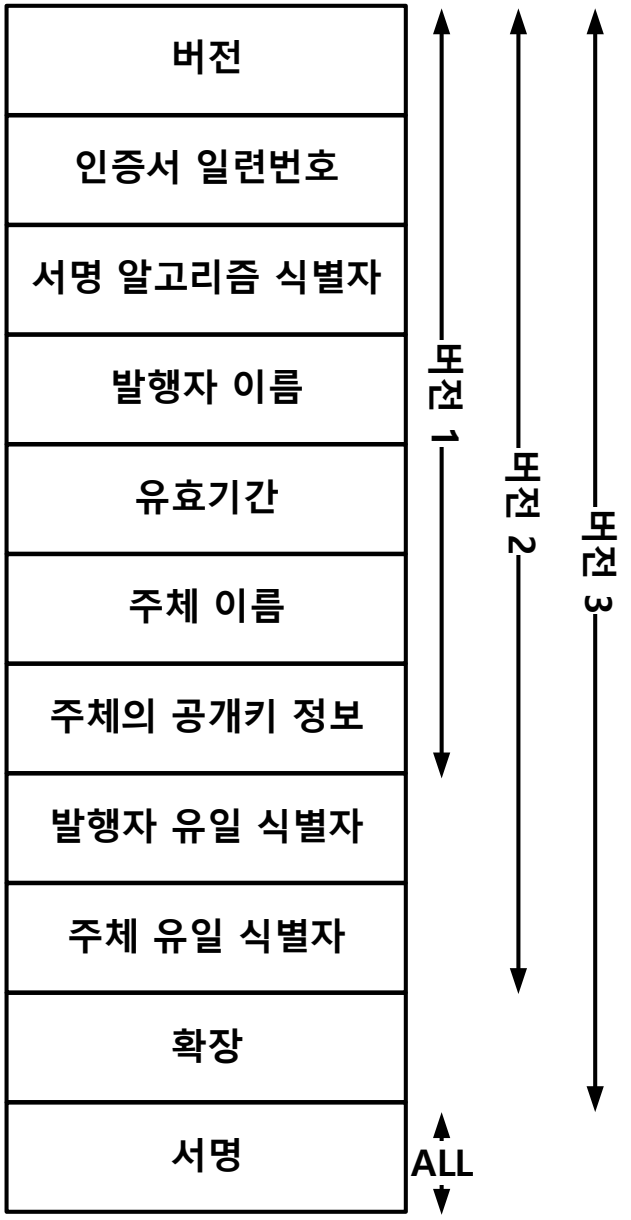
이름	의미
Issuer Unique ID	인증기관 식별 번호
Subject Unique ID	인증서 주체 식별 번호



X.509 인증서

- X.509 인증서 형식
  - X.509 v3의 확장 설명 표
    - 키와 정책 정보

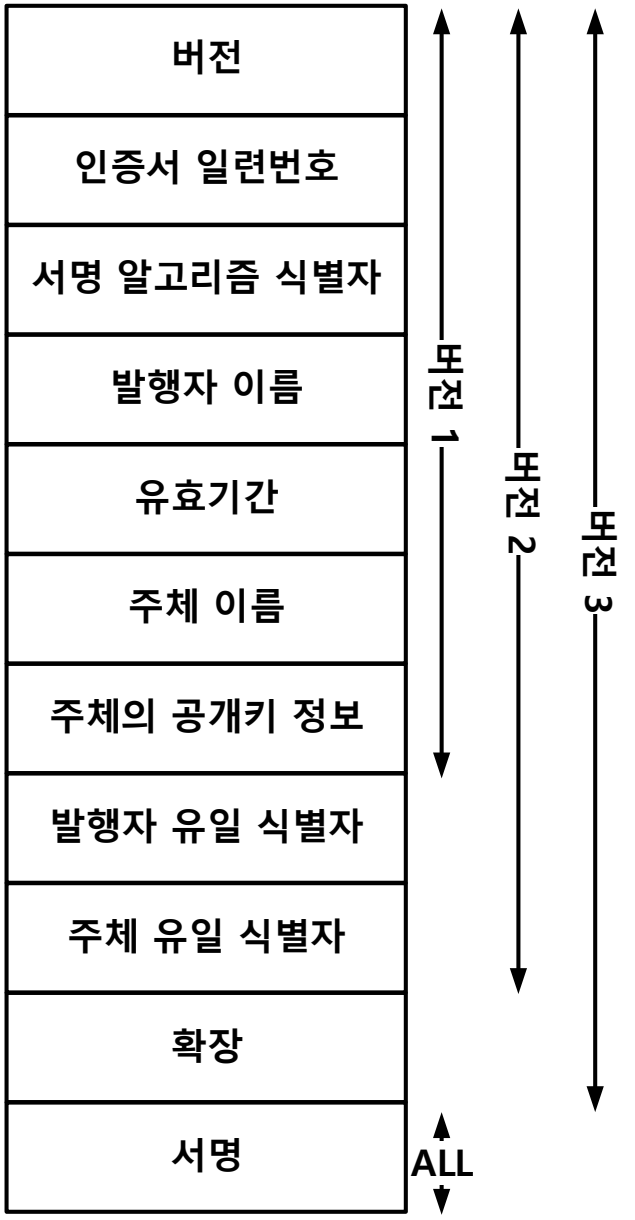
확장	의미
Authority Key ID	인증서, CRL에 있는 서명을 확인하는 공개키 식별
Subject Key ID	인증될 공개키를 식별
Key Usage	인증하는 공개키가 사용될 목적과 적용되는 정책에 따른 제한조건
Private-key Usage Period	인증하는 공개키에 대응하는 개인키의 사용 기간
Certificate Policies	해당 인증서가 지원하는 정책의 인식 항목
Policy Mapping	다른 CA에 의해 발행한 인증서에만 사용, 발행한 CA의 정책이 발행 받은 CA에 적용되는 다른 정책과 동일하게 간주됨을 나타냄



X.509 인증서

- X.509 인증서 형식
  - X.509 v3의 확장 설명 표
    - 인증서 주체와 발행자 속성

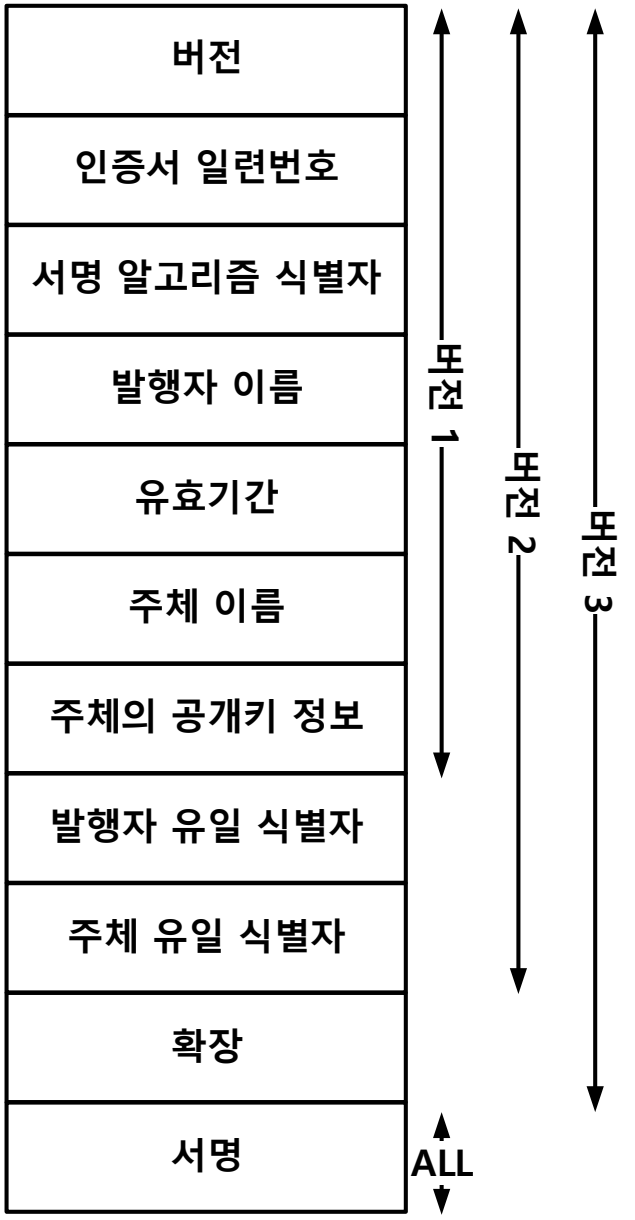
확장	의미
Subject Alternative Name	자체적으로 이름 형식을 가지고 있는 특정 응용프로그램을 지원하는 대체 이름 지원
Issuer Alternative Name	다양한 형식을 사용하는 하나 또는 여러 개의 대체이름 포함
Subject Directory Attributes	인증서의 주체를 위한 어떤 것이든 X.500 디렉터리 속성값을 표시



X.509 인증서

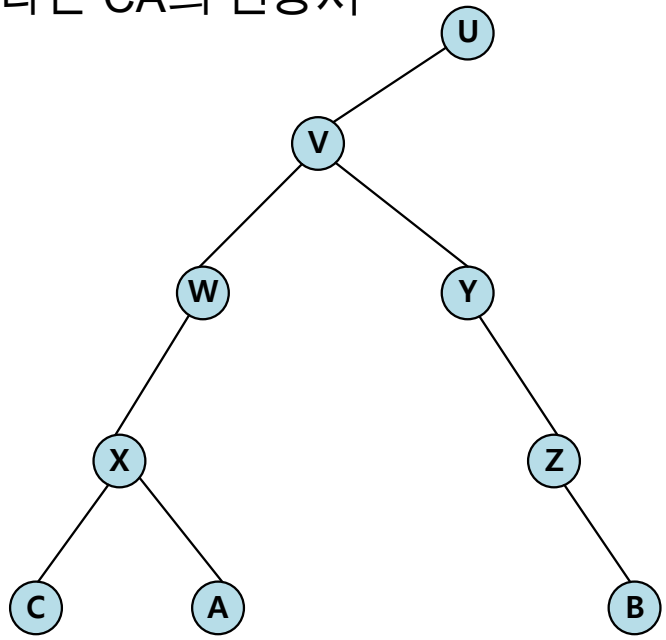
- X.509 인증서 형식
  - X.509 v3의 확장 설명 표
    - 인증경로 제약조건

확장	의미
Basic Constraints	주체가 CA역할인지 아닌지, 인증 경로에 대한 정보
Name Constraints	인증서에 나타나는 모든 주체의 이름이 위치해야 하는 이름 공간
Policy Constraints	구체적인 인증서 정책 식별 요구와 인증서 정책 매핑 금지 요구



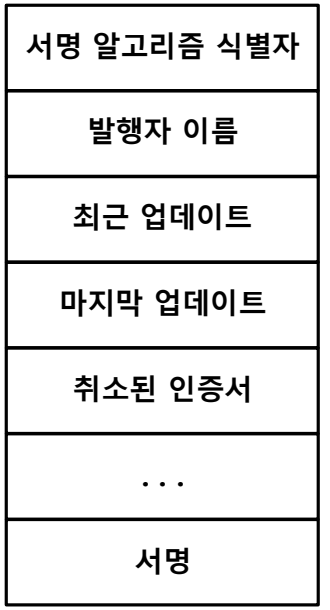
X.509 인증서

- X.509 타 기관 발행 인증서
  - X.509 발행하는 기관은 여러 개일 수 있고 서로 다른 기관으로부터 안전하게 공개키를 얻는 경로를 트리구조로 표현
    - Chain of Certificate
      - 각 CA의 디렉터리에는 두 종류의 인증서를 가지고 있어야함
        - 순방향 인증서(Forward Certificates): 다른 CA에 의해 생성된 X의 인증서
        - 역방향 인증서(Reverse Certificates): X가 생성한 다른 CA의 인증서
    - A에서 B에 이르는 인증 경로
      - X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>



X.509 인증서

- X.509 인증서의 취소
  - 각인증서는 유효기간이 포함됨
  - 유효기간 이외의 취소 사유
    - 사용자의 개인키 노출, 훼손
    - CA가 더 이상 사용자를 인증해줄 수 없음
    - CA의 인증서가 노출되었거나 훼손됨



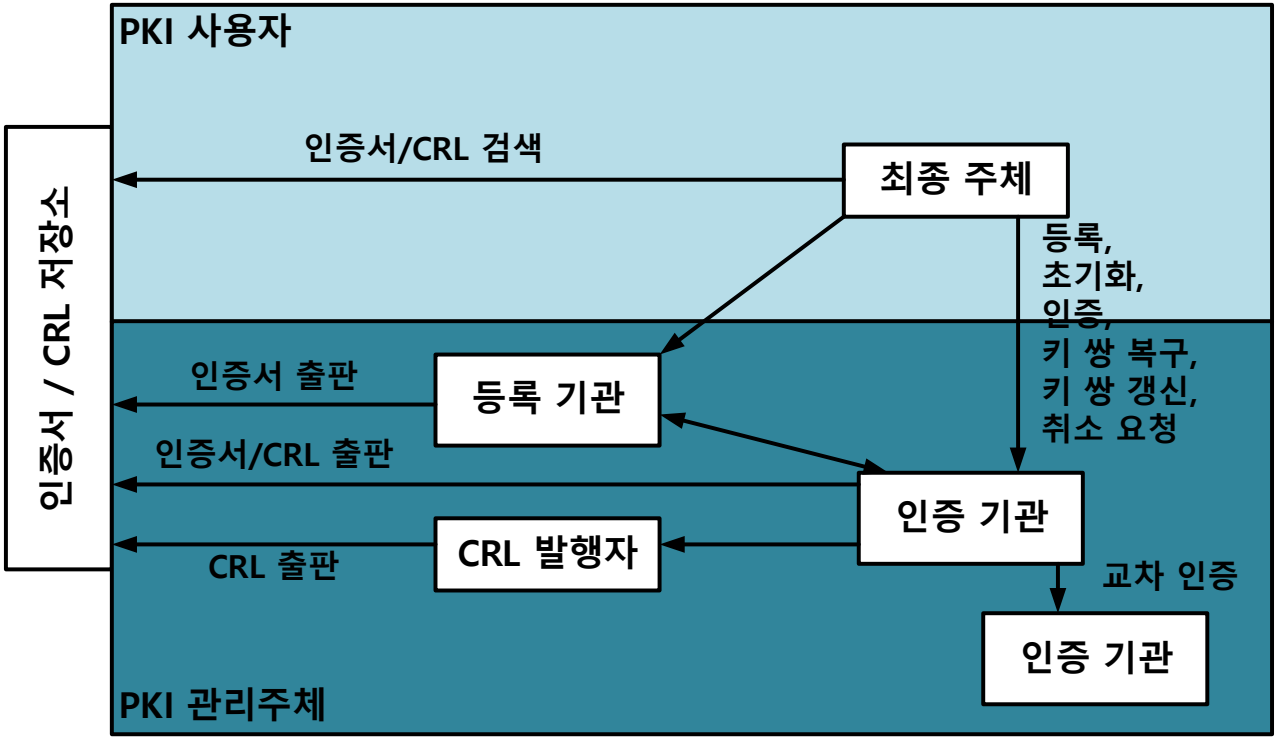
  - CA는 자신이 발행했지만 취소 되었고 유효기간이 아직 끝나지 않은 인증서의 목록을 관리하고 디렉터리에 공개
    - CRL(Certificate Revocation List)은 발행자에 의해 서명되어야 하고, 발행자 이름, 목록 작성일자, 다음 CRL발표 일자와 취소된 인증서 항목의 내용을 담아야 함

목 차

- 대칭 암호를 이용한 대칭키 분배
- Kerberos
- 비대칭 암호를 이용한 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합 신원 관리

공개키 기반구조

- 개요
  - RFC 2822(Internet Security Glossary)에서 공개키 기반 구조(PKI: Public-Key Infrastructure)는 비대칭 암호시스템에 기초한 디지털 인증 생성과 관리, 저장, 분배, 취소 에 필요한 하드웨어, 소프트웨어, 사람, 정책, 절차를 의미



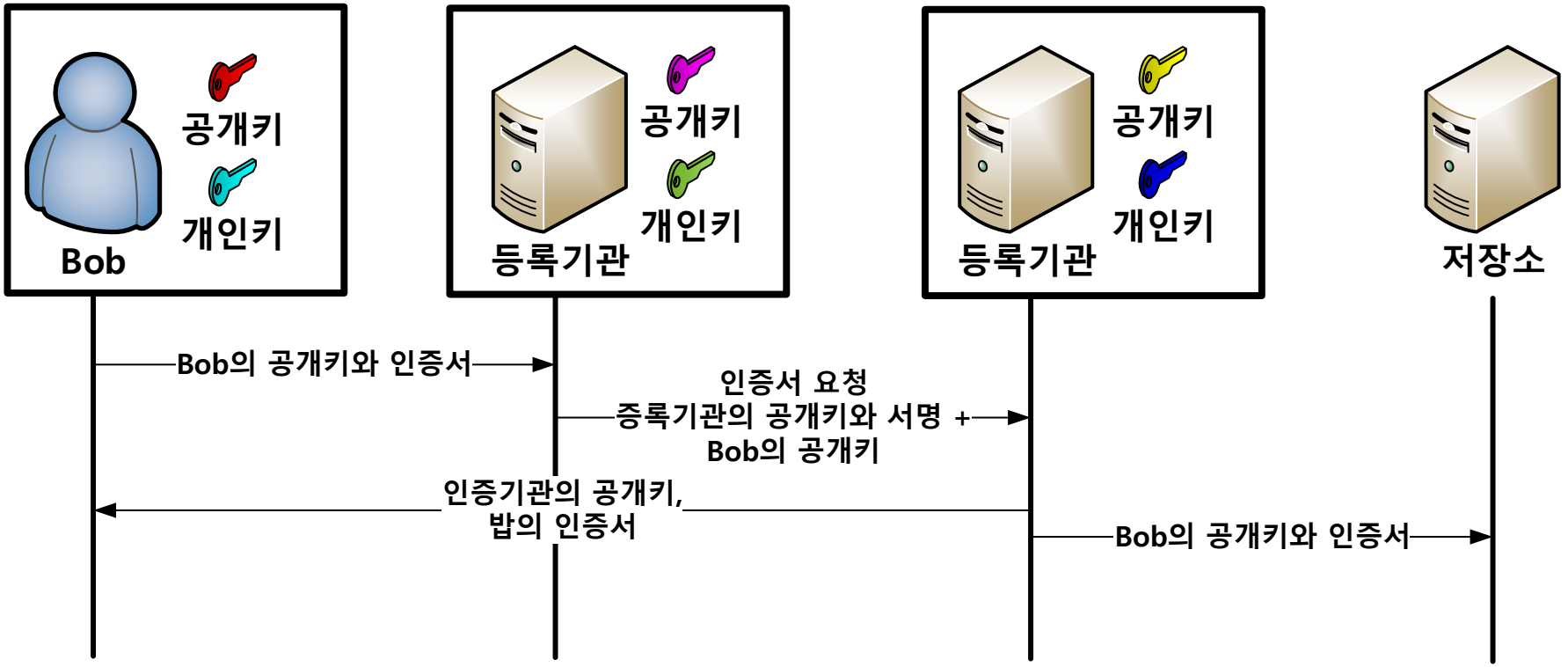
공개키 기반구조

- PKIX 관리 기능
  - 관리 프로토콜에 의해 지원하는 관리 규정 표

기능	역할
Registration	사용자가 최초로 CA에게 자신을 알림
Initialization	사용자 시스템을 안전하게 작동하기 위한 키 관련 재료 설치
Certificate	사용자에게 인증기관의 공개키와 인증서를 발급, 인증서를 저장
Key Pair Recovery	키가 오염되어 필요한 정보를 볼 수 없는 경우 복구하는 메커니즘 제공
Key Pair Update	모든 키 쌍을 정기적으로 갱신, 유효기간이 만료되거나 취소되는 경우
Revocation Request	권한을 가진 사람은 인증서 취소요청이 가능
Cross Certification	다른 CA와 교차 인증을 하기 위해 정보를 교환, A가 B에게 인증서 발급에 사용되는 CA의 서명키가 포함된 인증서를 발급

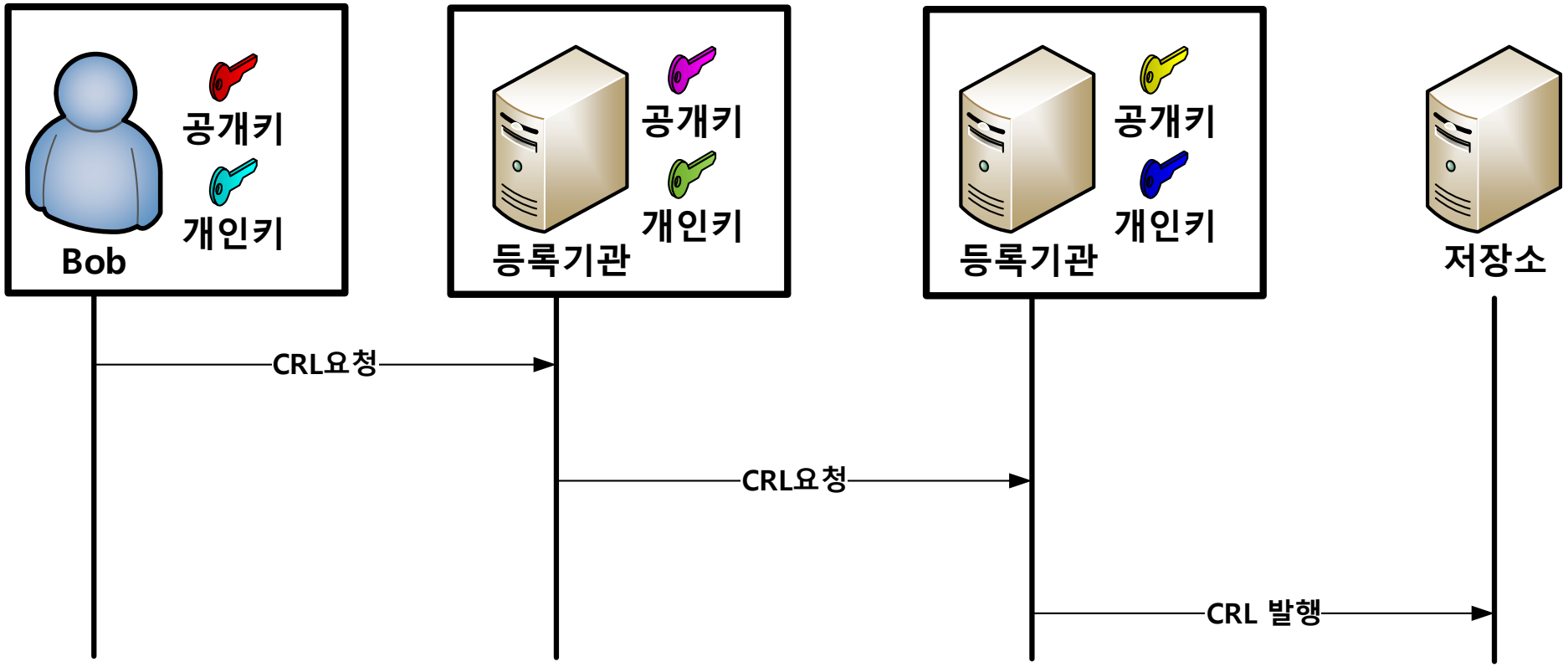
공개키 기반구조

- PKIX 관리 기능
  - 인증 과정



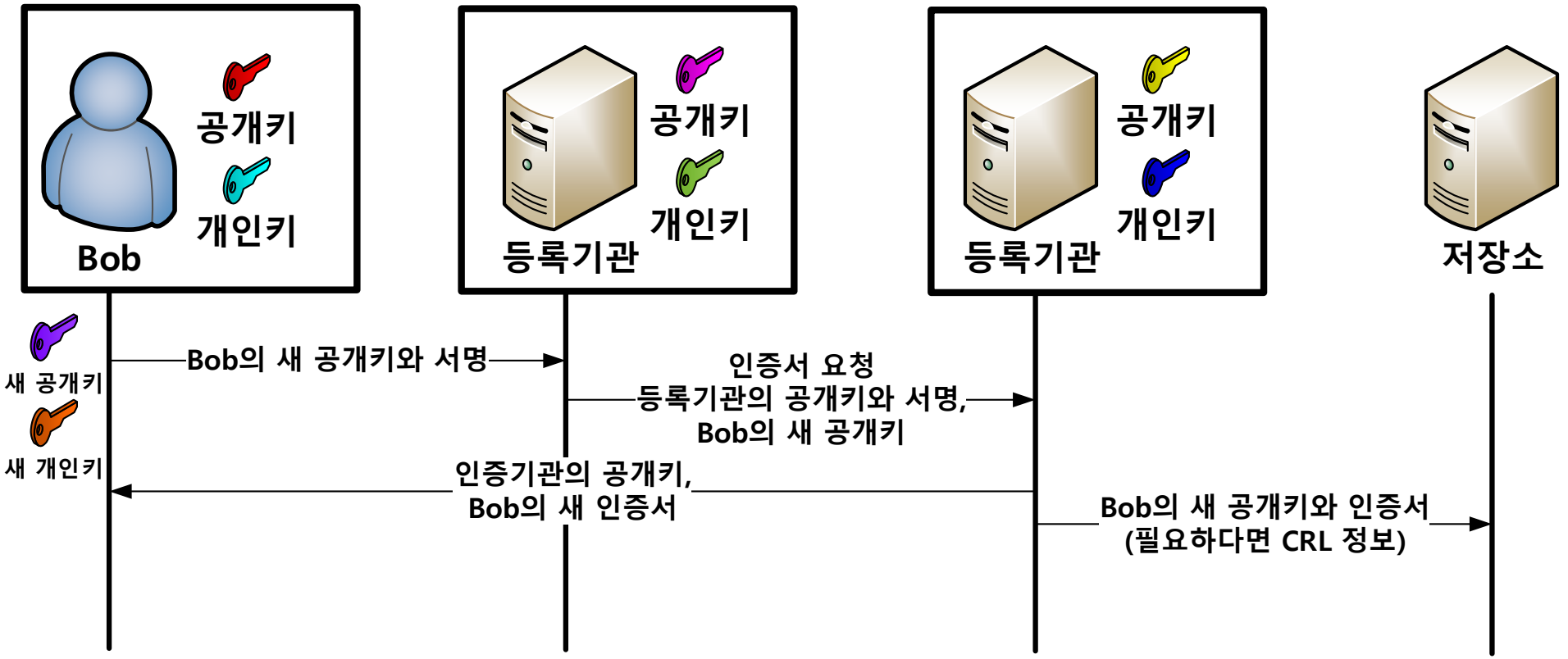
공개키 기반구조

- PKIX 관리 기능
  - 취소 요청 과정



공개키 기반구조

- PKIX 관리 기능
  - 키 쌍 갱신 과정



공개키 기반구조

- PKIX 관리 기능
  - 교차 인증 개념도

공개키 기반구조

- PKIX 관리 프로토콜
  - RFC 2510에서 인증서 관리 프로토콜 (CMP: Certificate Management Protocol)을 정의
  - RFC 2797은 CMS위에서 인증서 관리 메시지를 정의
    - RFC 2630 암호 메시지 구문 (CMS: Cryptographic Message Syntax)

목 차

- 대칭 암호를 이용한 대칭키 분배
- Kerberos
- 비대칭 암호를 이용한 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합 신원 관리

통합 신원 관리

- 개요
  - 통합 신원 관리(Federated Identity Management)는 다수의 기업과 여러 응용프로그램을 관리하는 신원 관리 시스템

- 신원 관리 (Identity Management)
  - 기업의 직원 또는 권한을 가진 개인이 자원에 접근하는 절차를 중앙 집중, 자동관리
  - 사용자 신원을 연관 짓고, 인증하는 방법을 강요
  - SSO(Single Sign-On)을 사용하여 한번 인증으로 네트워크의 모든 자원에 접근 가능하도록 함

통합 신원 관리

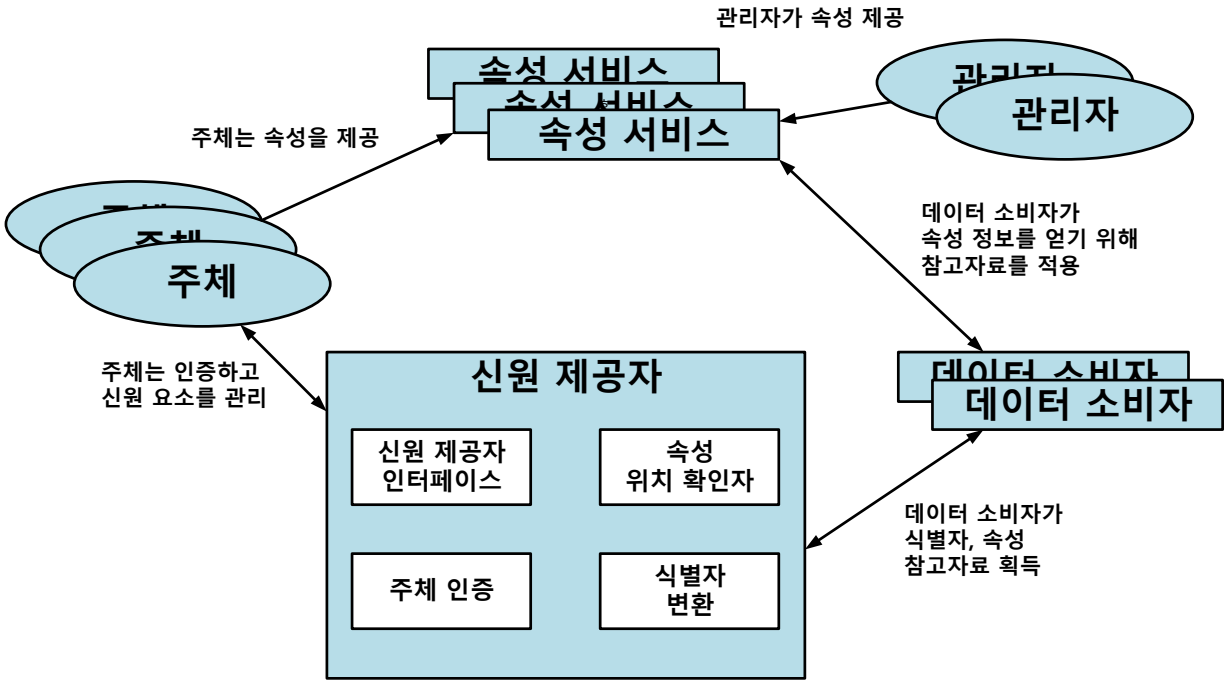
- 신원 관리 (Identity Management)
  - 신원 관리 원칙 목록 표

요소	의미
Authentication	제공한 사용자 이름과 사용자가 일치하는지 인증
Authorization	인증에 근거한 특정 서비스와 자원에 접근을 허가
Accounting	로그인 접근과 허가 절차
Provisioning	시스템에 사용자 등록하여 제공
Workflow Automation	비즈니스 프로세스에서 데이터의 이동
Delegated Administration (관리 위임)	허가 여부를 위한 역할-기반 접근 통제
Password Synchronization	SSO으로 일회 인증으로 네트워크 자원 접근
Self-Service PW Reset	사용자가 자신의 패스워드 갱신
Federation(통합)	인증과 허가 절차를 다른 시스템으로 전달, 매번 인증할 필요성 제거

통합 신원 관리

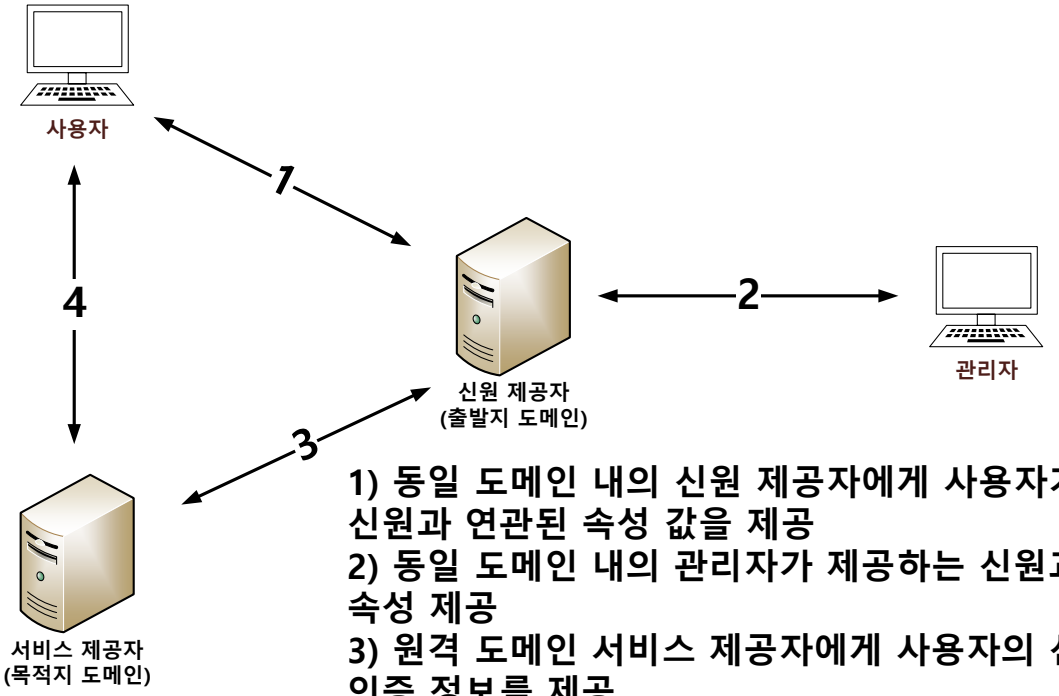
- 신원 관리 (Identity Management)
  - 신원 관리 의 일반 구조

주체 (Principal): 신원 소지자
신원 제공자 (Identity Provider): 인증정보를 주체와 연관
속성 서비스 (Attribute Service): 속성 정보를 생성/유지/관리
관리자 (Administrator): 사용자에게 속성을 부여
데이터 소비자(Data Consumers):
신원 제공자 또는 속성 서비스가 관리하는 데이터를 받아 사용하는 개체



통합 신원 관리

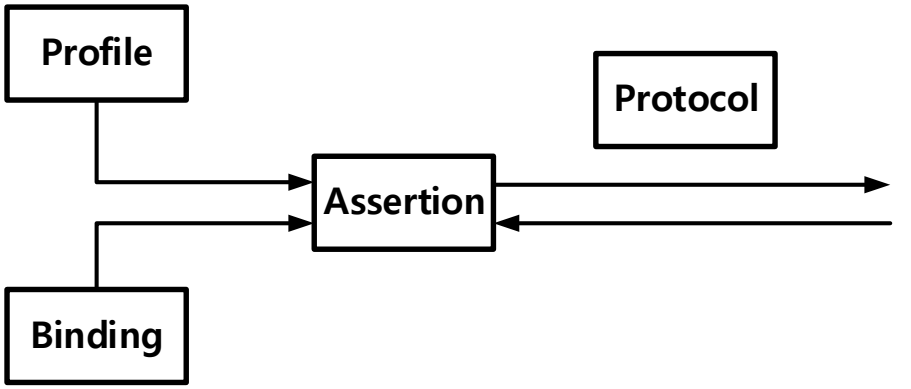
- 신원 통합 (Identity Federation)
  - 개요
    - 다른 도메인으로 신원 관리를 확장하고 신원을 공유하여 SSO으로 다수의 도메인에 있는 응용프로그램, 자원에 접근할 수 있도록 함



1) 동일 도메인 내의 신원 제공자에게 사용자가 사용자 신원과 연관된 속성 값을 제공
2) 동일 도메인 내의 관리자가 제공하는 신원과 연관된 속성 제공
3) 원격 도메인 서비스 제공자에게 사용자의 신원 정보와 인증 정보를 제공
4) 서비스 제공자는 원격 사용자와 세션을 생성하고 사용자의 신원 과 속성에 기반한 제한에 맞춰 접근 통제

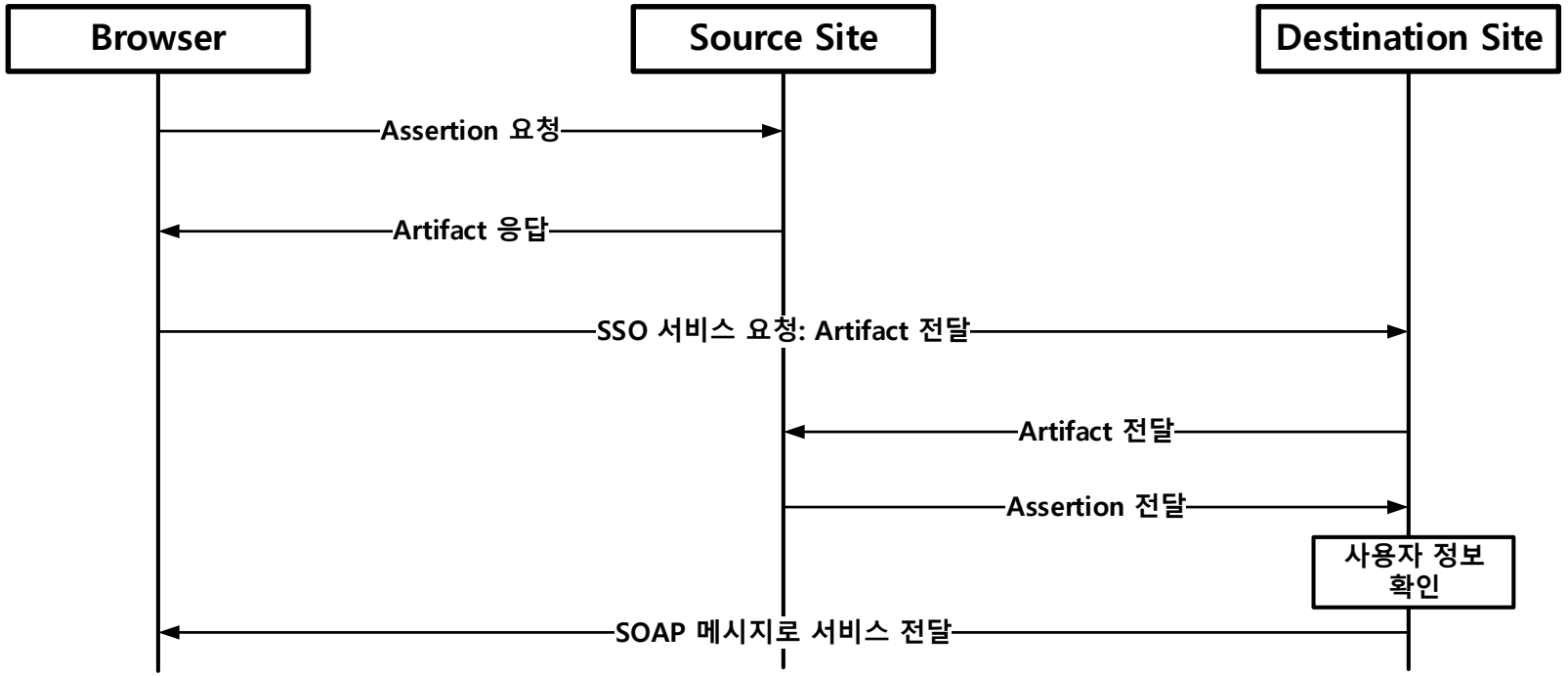
통합 신원 관리

- 신원 통합 (Identity Federation)
  - 신원 통합 구현
    - SAML(Security Assertion Markup Language)
      - 다양한 환경에서 ID와 인증 정보를 교류하기 위한 확장 가능한 데이터 포맷
      - 현대 네트워크 환경의 필수 조건



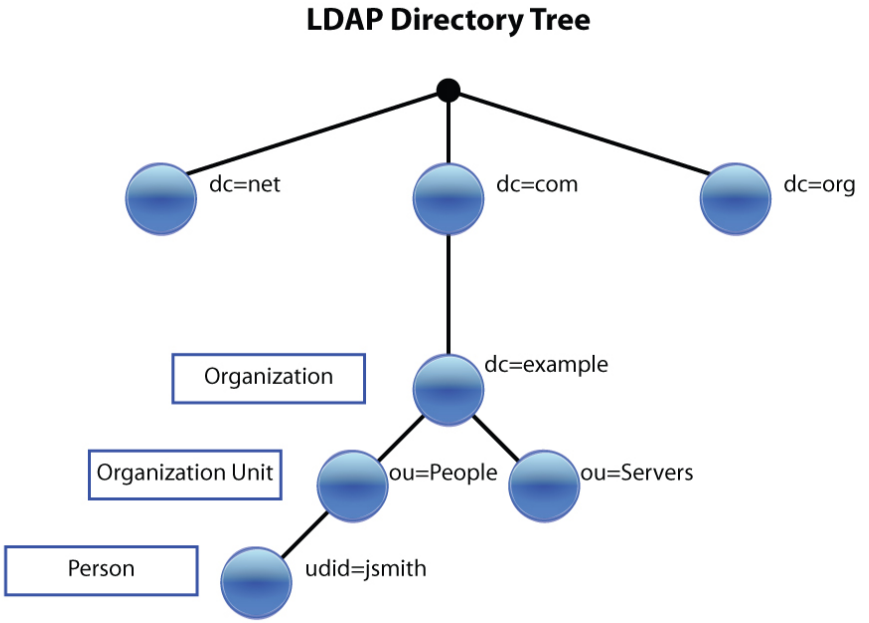
통합 신원 관리

- 신원 통합 (Identity Federation)
  - 신원 통합 구현
    - SAML(Security Assertion Markup Language) SSO 시나리오
      - SOAP(Simple object Access Protocol)
        - HTTP(80), SMTP(25) 등을 통해 XML기반의 메시지를 네트워크상에서 교환하는 프로토콜



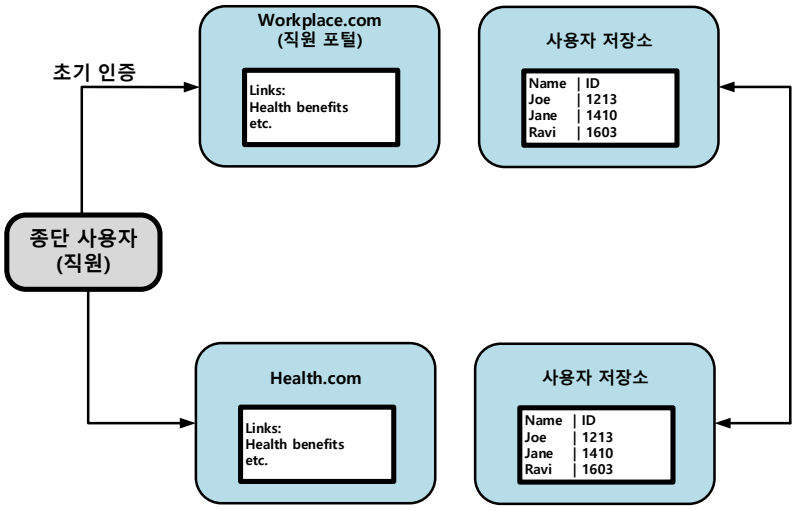
통합 신원 관리

- 신원 통합 (Identity Federation)
  - 신원 통합 구현
    - LDAP(Lightweight Directory Access Protocol)
      - 디렉터리 서비스 접근을 위한 클라이언트-서버 프로토콜 (TCP/389)

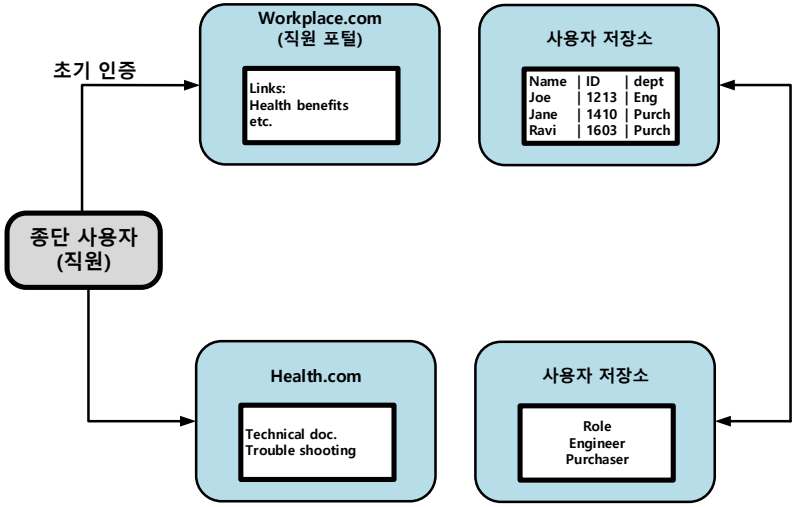


통합 신원 관리

- 신원 통합 (Identity Federation)
  - 신원 통합의 예
    - 계정 연결기반 통합
      - 인증된 계정이 외부 서비스에 접근하면 전달 받은 식별자로 인증
    - 역할 기반 통합
      - 인증시 역할 구분에 따라 외부 서비스에 접근할 수 있는 권한이 부여됨



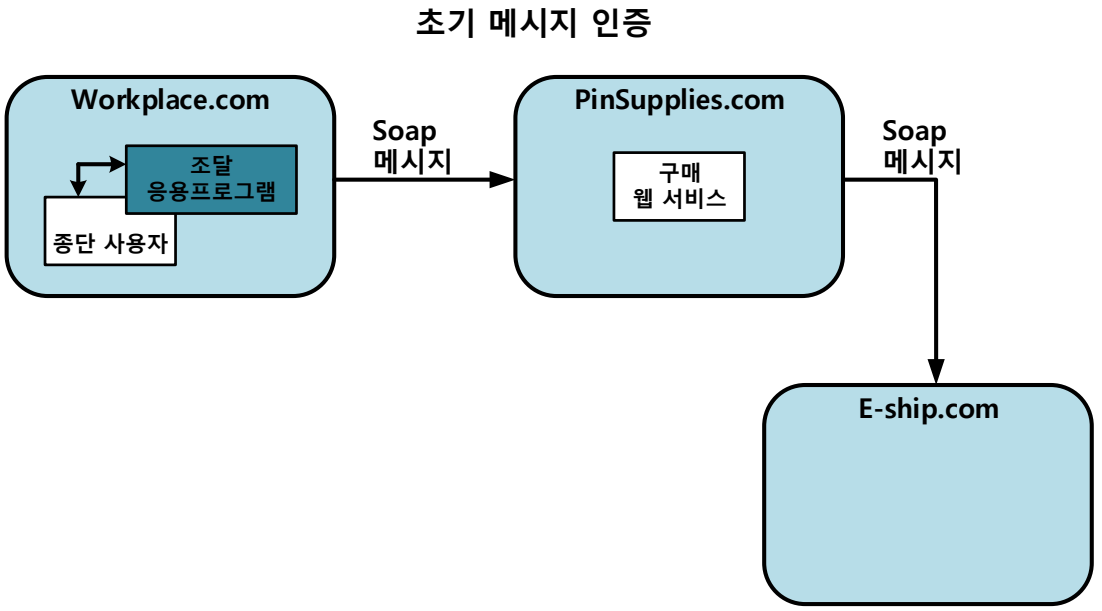
계정 연결기반 통합



역할 기반 통합

통합 신원 관리

- 신원 통합 (Identity Federation)
  - 신원 통합의 예
    - 연쇄 웹 서비스
      - 인증하고 외부 서비스 요청시 전송하는 문서, 파일 기반으로 인증이 되고, 외부 서비스 이용



감사합니다!