2021/03/05, 2021 보안 기초 세미나

Network Security Essentials

- Chapter 4 키 분배와 사용자 인증 -

박 재 형(jaehyoung@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

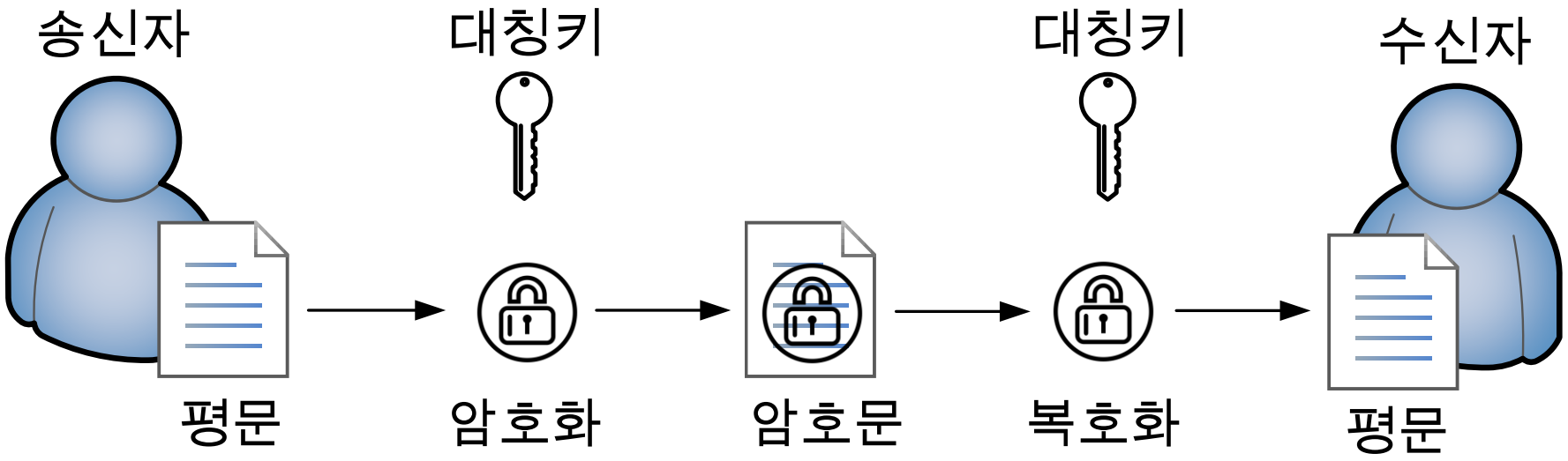
- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

목 차

- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

대칭키 암호의 키 분배

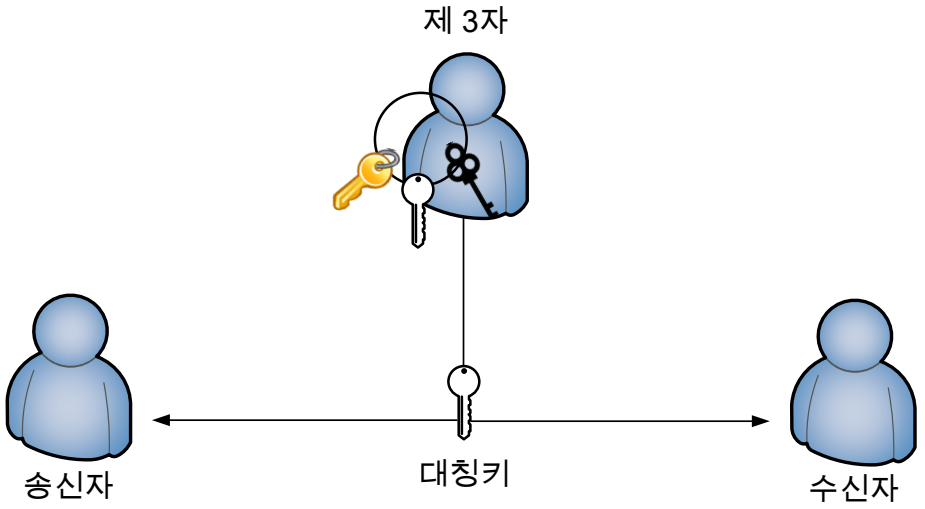
- 대칭키 암호
  - 정의
    - 동일한 키를 사용하여 평문을 암호화 하는 기법
      - 대칭키
  - 대칭키를 분실 및 유출 시, 원하지 않는 사용자가 평문 내용을 볼 수 있음



대칭키 암호의 키 분배

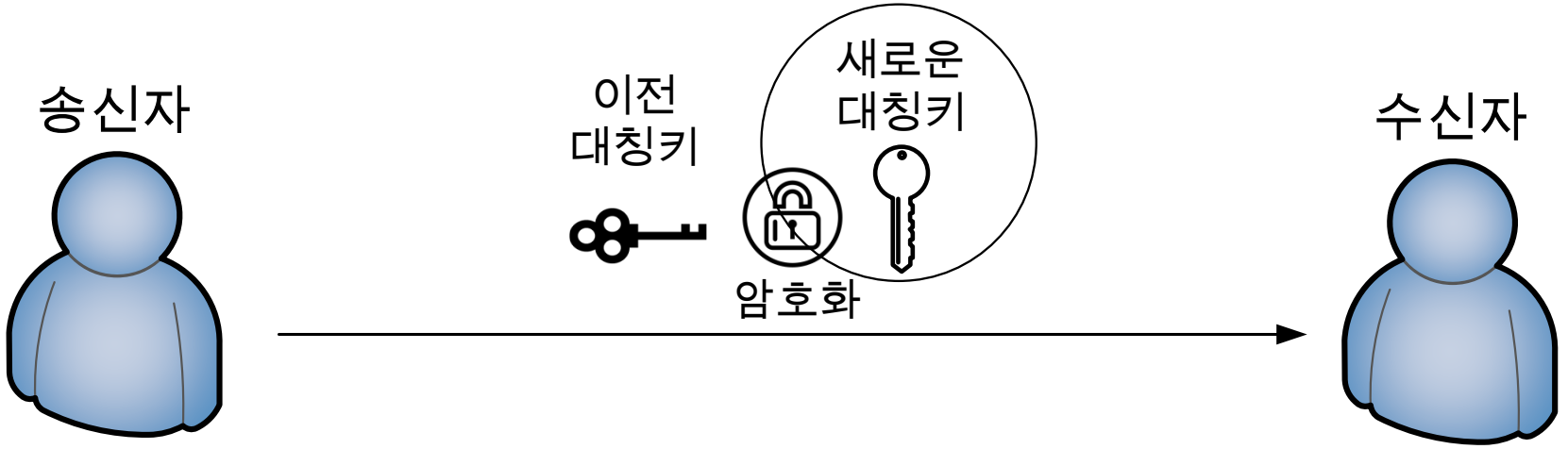
- 대칭키 암호
  - 키 교환 방법(1/3)
    - 송신자가 키를 선택한 뒤, 수신자에게 직접 전달

    - 제 3자가 키를 선택한 뒤, 송신자와 수신자에게 직접 전달



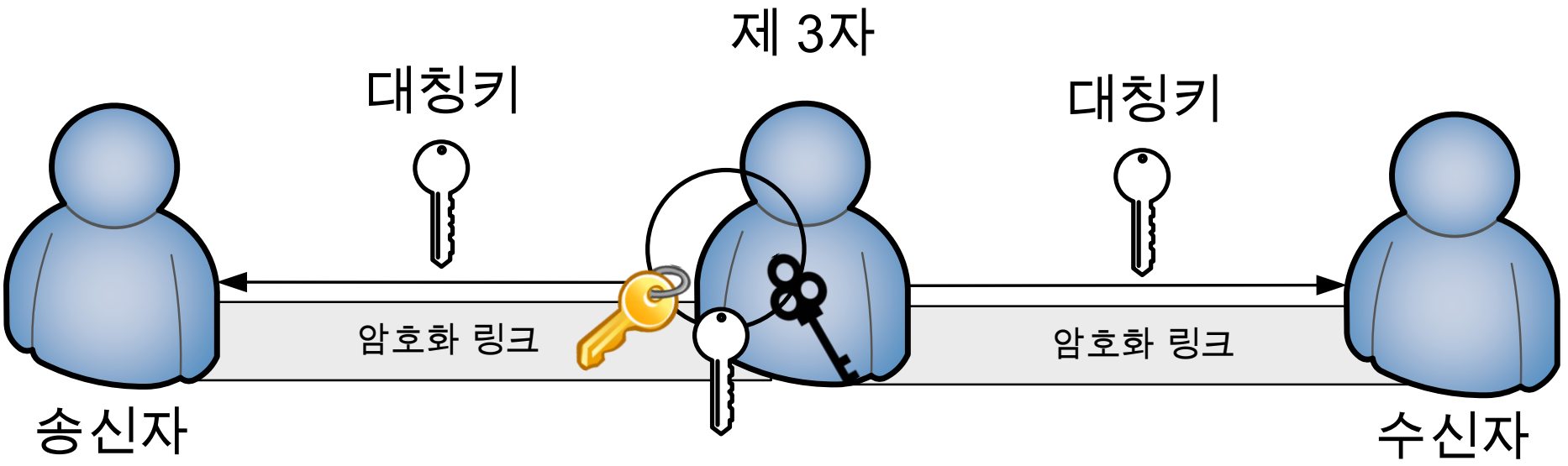
대칭키 암호의 키 분배

- 대칭키 암호
  - 키 교환 방법(2/3)
    - 송신자와 수신자가 이전에 사용한 키가 있다면, 둘 중 한사람이 새로운 키를 만들고 이전 키로 암호화하여 전달



대칭키 암호의 키 분배

- 대칭키 암호
  - 키 교환 방법(3/3)
    - 송신자와 수신자가 제 3자와 암호화 연결이 확립된 경우, 제 3자가 암호화된 링크를 통해 송신자와 수신자에게 키 전달



대칭키 암호의 키 분배

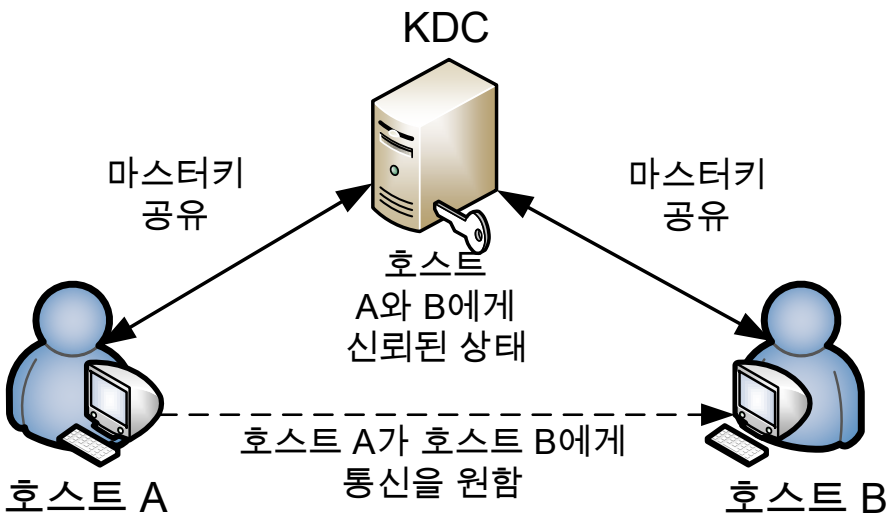
- 대칭키 암호
  - 용어

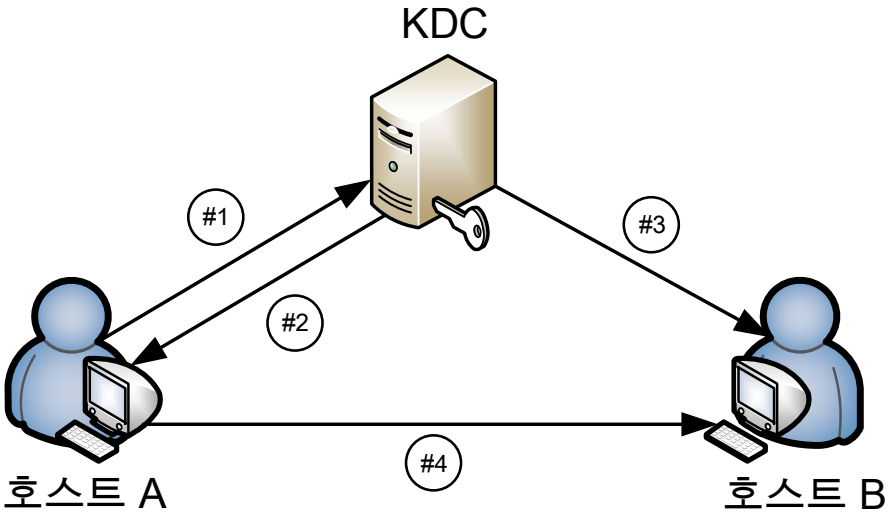
용어	정의	설명
세션키 (Session Key)	통신에서 메시지를 암호화하기 위해 사용되는 일회용 대칭키	• 두 개의 종단 시스템이 통신하기 위한 논리적 연결이 구성되어야 함 • 세션이 유지되는 동안 모든 사용자 데이터는 세션키로 암호화 • 세션이 종료되면 세션키는 폐기
마스터키 (Master Key)	KDC와 개체 간의 공유하고 있는 키	• 개체에게 세션키를 분배하기 위해 사용
키분배센터 (KDC, Key Distribution Center)	통신 당사자가 아닌 신뢰된 제 3자	• 연결하려는 두 개체가 인증되면 일회용 세션 키를 양쪽 개체에게 제공

대칭키 암호의 키 분배

- 키분배센터(KDC, Key Distribution Center)
  - 동작과정

순서	설명
#1	• 호스트 A가 호스트 B에게 연결할 경우, 연결 요청 패킷을 KDC에 전송 • 호스트A와 KDC가 공유하고 있는 마스터키를 이용해서 암호화 • 서비스 이용 여부 인증을 위함
#2	• KDC가 연결 요청 허락 시, 일회용 세션키 생성 및 전달 • 호스트 A와 KDC가 공유하고 있는 마스터키를 이용해서 암호화
#3	• KDC가 호스트 B에게 연결을 요청하며 동의할 경우, 세션키 전달 • 호스트 B와 KDC가 공유하고 있는 마스터키를 이용해서 암호화
#4	• 호스트 A와 호스트 B 통신 • 세션키를 이용하여 메시지를 암호화





대칭키 암호의 키 분배

- 키분배센터(KDC, Key Distribution Center)
  - 문제점
    - 개체가 증가할 경우, KDC에서 보관하는 키의 개수 증가
      - 개체가 n명 일 경우, 키의 개수는 $n(n+1)/2$
    - KDC에서 모든 키를 관리하기 때문에 공격당할 경우, 모든 사용자의 키 노출

목 차

- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

Kerberos

- 개요
  - MIT(Massachusetts Institute of Technology)에서 개발한 대칭키 암호 방식의 키 분배 및 사용자 인증 기법
  - 서버의 워크스테이션 구별 위협
    - 공격자가 워크스테이션에 접근한 경우, 그 워크스테이션에서 작업하는 사용자로 가장할 수 있음
    - 공격자가 워크스테이션 네트워크 주소를 변경한 경우, 송신한 요청사항이 정상 워크스테이션에서 요청한 것처럼 가장할 수 있음
    - 공격자가 통신을 도청하고 이를 재전송한 경우, 서버에 대한 접근 허락을 받거나 통신과정을 방해할 수 있음

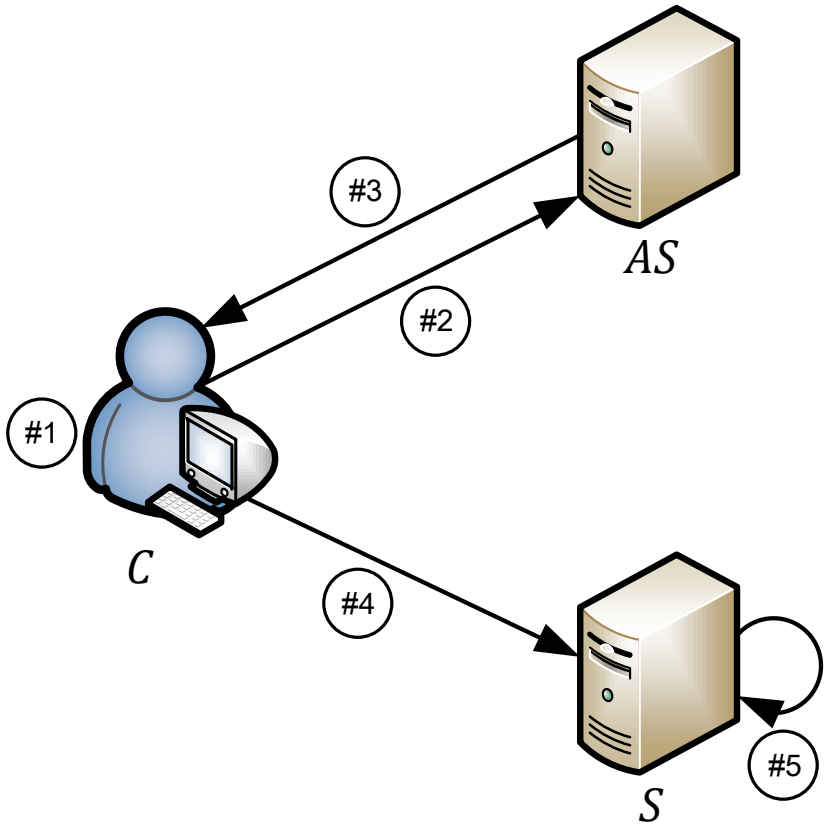
Kerberos

- 특징
  - 서버는 서비스를 이용하려고 하는 사용자를 인증
  - 사용자는 서버를 인증
  - 중앙집중식 인증 서버 구축
  - 대칭키 암호 사용
- 용어

용어	설명
C	클라이언트
AS	인증 서버
S	서버
ID_c	C 안에서 사용되는 사용자 식별자
ID_S	S의 식별자
PW_c	C 안에서 사용되는 사용자 패스워드
AD_c	C 의 네트워크 주소
K_S	AS 와 S 가 공유하는 비밀 암호화 키
$Ticket$	사용자 ID, 네트워크 주소, 서버 ID를 포함하는 암호화된 메시지
$\|\|$	메시지 이어 붙이기

Kerberos

- Kerberos 버전 4
  - 단순 인증 절차
    - 동작 과정 그림



#1 사용자 로그인

#2 $ID_C \,||\, ID_S \,||\, PW_C$ 전송

#3 ID_C와 PW_C 그리고 서버 접근 여부 확인 후,
$Ticket = E(K_S, [ID_C \,||\, AD_C \,||\, ID_S])$ 전송

#4 $ID_C \,||\, Ticket$ 전송

#5 $Ticket$을 복호화 하여 ID_C를 비교한 후,
서비스 승인

Kerberos

- Kerberos 버전 4
  - 단순 인증 절차
    - 문제점
      - 사용자가 입력해야 할 패스워드의 횟수 증가
        - 새로운 서비스를 이용할 때마다 새로운 $Ticket$을 발행 받기 위해 패스워드를 입력해야 함
      - 평문으로 전달되는 패스워드
        - 공격자는 패스워드를 가로채 해당 사용자에게 허용된 모든 서비스를 이용가능

Kerberos

- TGS(Ticket-Granting Server)
  - 정의
    - 사용자와 서버 간의 통신을 하기위한 $Ticket$을 발급하는 서버
  - 특징
    - 패스워드를 평문으로 사용하지 않음
    - AS에게 인증 받은 사용자에게 $Ticket$ 발급
    - 요청된 서비스 별로 $Ticket$ 발급

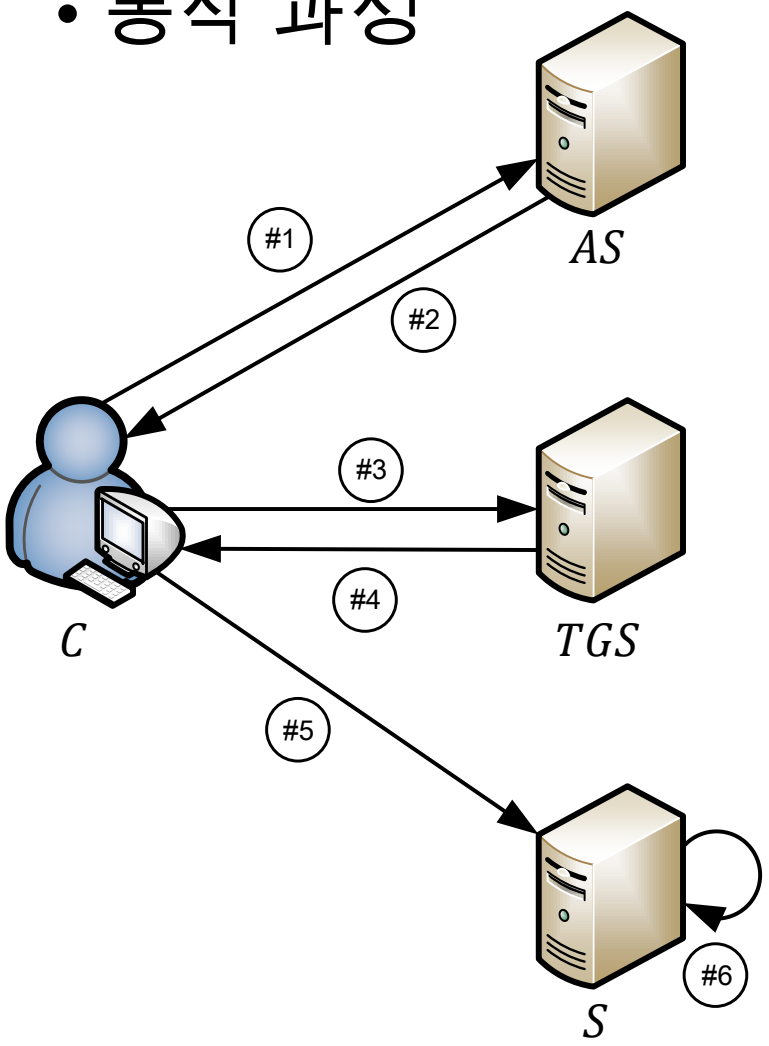
Kerberos

- TGS를 도입한 인증 절차
  - 용어

용어	설명
C	클라이언트
AS	인증 서버
S	서버
TGS	티켓 발행 서버
ID_c	C 클라이언트의 식별자
ID_S	서버의 식별자
ID_{TGS}	$Ticket$ 발행 서버의 식별자
PW_c	클라이언트의 패스워드
AD_c	클라이언트의 IP 주소
K_C	클라이언트의 패스워드로 부터 얻은 키
K_{AS-TGS}	인증 서버와 $Ticket$ 발행 서버가 공유하는 비밀키
K_{TGS-S}	$Ticket$ 발행 서버와 서버가 공유하는 비밀키
TS	타임스탬프
$Lifetime$	$Ticket$ 유효기간
$\|\|$	메시지 이어 붙이기

Kerberos

- TGS를 도입한 인증 절차
  - 동작 과정



인증 서비스 교환: *Ticket*-발행 *Ticket*을 취득

#1 $ID_C \,||\, ID_{TGS}$ 전송

#2 $E(K_C, Ticket_{TGS})$ 전송

- $Ticket_{TGS} = E(K_{AS-TGS}, [ID_C \,||\, AD_C \,||\, ID_{TGS} \,||\, TS_1 \,||\, Lifetime_1])$

***Ticket*-발행 서비스 교환: 서비스-발행 *Ticket* 취득**

#3 사용자로부터 패스워드를 입력 받은 후, K_C를 생성하여 메시지 복호화

$ID_C \,||\, ID_S \,||\, Ticket$ 전송

#4 $Ticket_{TGS}$을 복호화하여 ID_{TGS} 존재여부, 유효기간 만료여부, ID_C, PW_C, 서버 접근 여부를 확인 후, $Ticket_S$ 전송

- $Ticket_S = E(K_{TGS-S}, [ID_C \,||\, AD_C \,||\, ID_S \,||\, TS_2 \,||\, Lifetime_2])$

클라이언트/서버 인증 교환: 서비스 취득

#5 $ID_S \,||\, Ticket_S$ 전송

#6 $Ticket_S$을 복호화하여 ID_S를 비교한 후, 서비스 승인

Kerberos

- TGS를 도입한 인증 절차
  - 문제점
    - $Ticket$ 의 유효기간
      - $Ticket$ 의 유효기간이 짧을 경우, 사용자는 세션이 끝나지 않았음에도 불구하고 패스워드를 입력해야 하는 상황 발생
      - $Ticket$ 의 유효기간이 길 경우, 사용자가 서비스 사용을 종료한 후에도 $Ticket$ 은 사용이 가능하여 공격자가 유효기간이 남은 $Ticket$ 을 이용
    - 서버 인증
      - 공격자가 서버의 식별자를 위조하여, 가짜 서버를 이용해 서비스를 제공할 수 있음
        - 사용자는 서비스를 이용하고 있다고 착각
        - 사용자가 서비스를 제공받지 못하도록 함

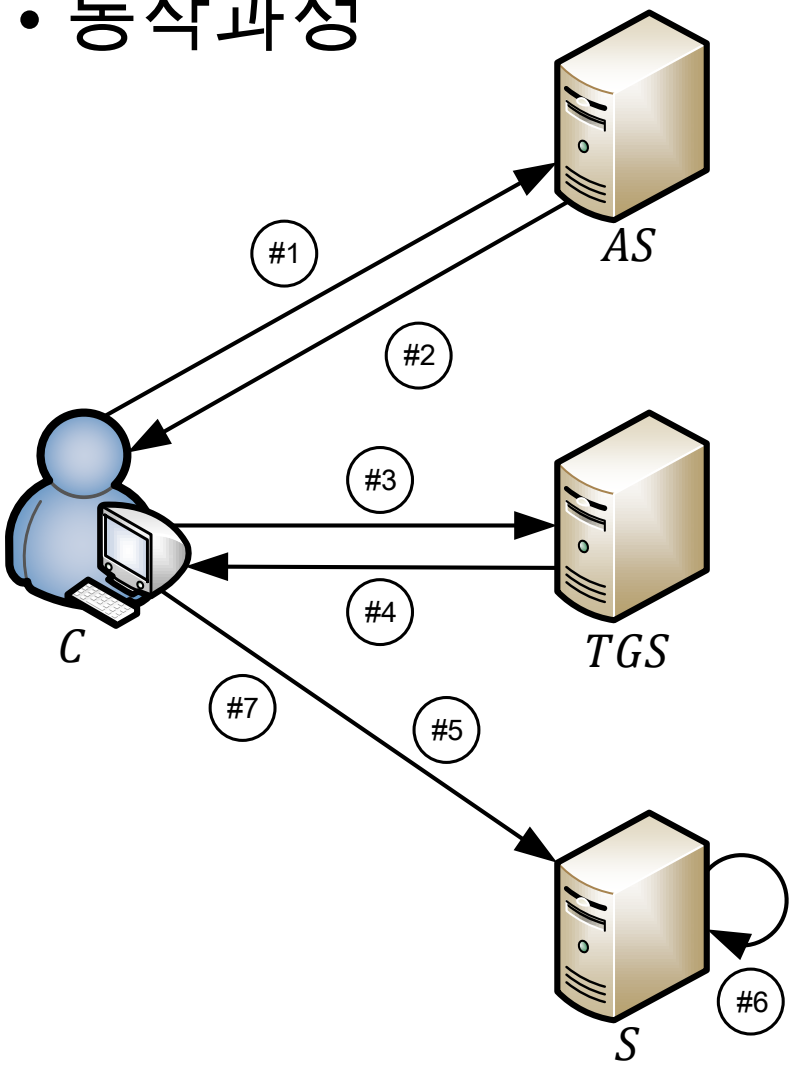
Kerberos

- Kerberos 버전 4
  - 용어

용어	설명
C	클라이언트
AS	인증 서버
S	서버
TGS	티켓 발행 서버
ID_c	C 클라이언트의 식별자
ID_S	서버의 식별자
ID_{TGS}	$Ticket$ 발행 서버의 식별자
PW_c	클라이언트의 패스워드
AD_c	클라이언트의 IP 주소
K_C	클라이언트의 패스워드로 부터 얻은 키
K_{C-TGS}	클라이언트와 서버 사이의 세션키
K_{C-S}	클라이언트와 서버 사이의 세션키
K_{AS-TGS}	인증 서버와 $Ticket$ 발행 서버가 공유하는 비밀키
K_{TGS-S}	$Ticket$ 발행서버와 서버가 공유하는 비밀키
TS	타임스탬프
$Lifetime$	$Ticket$ 유효기간
$\|\|$	메시지 이어 붙이기

Kerberos

- Kerberos 버전 4
  - 동작과정



인증 서비스 교환: *Ticket*-발행 *Ticket*을 취득

#1 $ID_C \,||\, ID_{TGS} \,||\, TS_1$ 전송

#2 $E(K_{C-TGS} \,||\, ID_{TGS} \,||\, TGS_2 \,||\, Lifetime \,||\, Ticket_{TGS})$ 전송

- $Ticket_{TGS} = E(K_{AS-TGS}, [K_{C-TGS} \,||\, ID_C \,||\, AD_C \,||\, ID_{TGS} \,||\, TS_2 \,||\, Lifetime_2])$

***Ticket*-발행 서비스 교환: 서비스-발행 *Ticket* 취득**

#3 사용자로부터 패스워드를 입력 받은 후, K_C를 생성하여 메시지를 복호화 이후, $ID_S \,||\, Ticket_{TGS} \,||\, Authenticator_{C-TGS}$ 전송

- $Authenticator_{C-TGS} = E(K_{C-TGS}, [ID_C \,||\, AD_C \,||\, ID_S \,||\, TS_4 \,||\, Lifetime_4])$

#4 $Ticket_{TGS}$과 $Authenticator_{C-TGS}$를 복호화 후, ID_C와 AD_C가 일치할 경우, $E(K_{C-TGS}, [K_{C-S} \,||\, ID_S \,||\, TS_4 \,||\, Ticket_S])$ 전송

- $Ticket_S = E(K_{TGS-S}, [K_{C-S} \,||\, ID_C \,||\, AD_C \,||\, ID_S \,||\, TS_4 \,||\, Lifetime_4])$

클라이언트/서버 인증 교환: 서비스 취득

#5 메시지를 복호화하여 $Ticket_S \,||\, Authenticator_{C-S}$ 전송

- $Authenticator_{C-S} = E(K_{C-S}, [ID_C \,||\, AD_C \,||\, TS_5])$

#6 $Ticket_S$과 $Authenticator_{C-S}$를 복호화 후, 서비스 승인

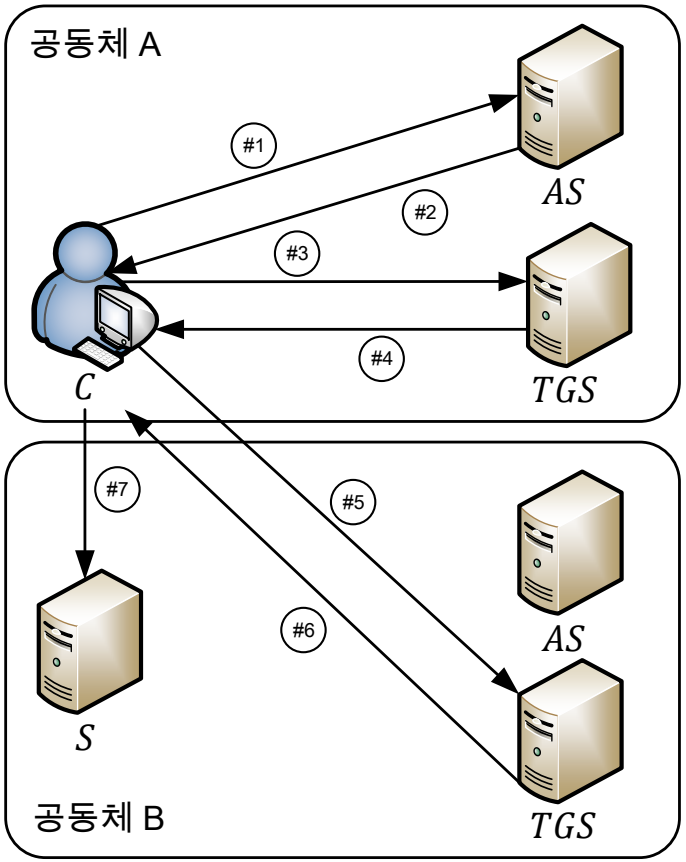
#7 상호간의 인증이 필요한 경우, $E(K_{C-S}, [TS_5 + 1])$ 전송

Kerberos

- Kerberos 버전 4
  - 구성 환경
    - Kerberos 서버, 다수의 클라이언트, 다수의 응용 서버로 구성된 환경
      - Kerberos 서버는 모든 사용자의 ID와 해시된 패스워드를 데이터베이스에 저장하고 있어야함
      - Kerberos 서버는 각 서버와 비밀키를 공유해야 함
        - Kerberos 공동체(Kerberos realm)
      - Kerberos 서버는 상호 교류하는 서로 다른 공동체 서버와 비밀키를 공유해야 함

Kerberos

- Kerberos 버전 4
  - Kerberos 공동체 사이의 인증
    - 동작과정



순번	동작
#1	• 로컬 TGS용 $Ticket$ 요청 • $ID_C \| ID_{TGS} \| TS_1$
#2	• 로컬 TGS용 $Ticket$ • $E(K_C, [K_{C-TGS} \| ID_{TGS} \| TS_2 \| Lifetime_2 \| Ticket_{TGS}])$
#3	• 원격 TGS용 $Ticket$ 요청 • $ID_{REMTGS} \| Ticket_{TGS} \| Authenticator_C$
#4	• 원격 TGS용 $Ticket$ • $E(K_{C-TGS}, [K_{C-REMTGS} \| ID_{REMTGS} \| TS_4 \| Ticket_{REMTGS}])$
#5	• 원격 서버용 $Ticket$ 요청 • $ID_{REMS} \| Ticket_{REMTGS} \| Authenticator_C$
#6	• 원격 서버용 $Ticket$ • $E(K_{C-REMTGS}, [K_{C-REMS} \| ID_{REMS} \| TS_6 \| Ticket_{REMS}])$
#7	• 원격 서비스 요청 • $Ticket_{REMS} \| Authenticator_C$

Kerberos

- Kerberos 버전 5
  - Kerberos 버전 4의 환경적 결함 보완

결함	보완
DES암호화 알고리즘을 사용해야함	• 모든 종류의 암호기술 사용 가능 • e.g., AES, CRC, RSA
IP 주소를 사용해야함	• 어떤 유형의 네트워크 주소도 사용가능 • e.g., 포트, MAC
티켓의 유효기간은 최대 1280분으로 장시간 실행되는 응용 프로그램 같은 경우에 부적합	• 티켓에 시작시간과 만료시간을 명시
클라이언트가 사용하는 인증자는 한 가지 서버에만 사용 가능	• 서로 다른 서버에서도 인증자 사용 가능 • 인증자: C의 정보를 S와 공유한 세션키로 암호화한 메시지
n개의 공동체가 다른 모든 Kerberos 공동체와 상호 교류를 위해서는 $n(n+1)/2$번의 키 교환 필요	• 이보다 적은 숫자의 관계 요구

Kerberos

- Kerberos 버전 5
  - Kerberos 버전 4의 기술적 결함 보완

결함	보완
$Ticket$ 이중 암호화로 인한 낭비	• 모든 종류의 암호기술 사용 가능 • e.g., AES, CRC, RSA
PCPB 비표준 모드의 DES 사용으로 인한 암호문 블록을 교환하는 공격 기법에 취약	• CBC 모드 사용으로 무결성 제공
서비스 사용에 동일한 티켓이 반복적으로 사용될 수 있어 공격자가 이전 세션에서 가져온 메시지를 통해 재전송 공격 가능	• 클라이언트와 서버가 협상할 수 있도록 도입한 서브 세션키를 사용하여 클라이언트가 접근을 시도할 때마다 서브 세션키를 생성

Kerberos

- Kerberos 버전 5
  - 용어

용어	설명
C	클라이언트
AS	인증 서버
S	서버
TGS	티켓 발행 서버
ID_c	C 클라이언트의 식별자
ID_S	서버의 식별자
ID_{TGS}	$Ticket$ 발행 서버의 식별자
PW_c	클라이언트의 패스워드
AD_c	클라이언트의 IP 주소
K_C	클라이언트의 패스워드로 부터 얻은 키
K_{C-TGS}	클라이언트와 서버 사이의 세션키
K_{C-S}	클라이언트와 서버 사이의 세션키
K_{AS-TGS}	인증 서버와 $Ticket$ 발행 서버가 공유하는 비밀키
K_{TGS-S}	$Ticket$ 발행서버와 서버가 공유하는 비밀키

용어	설명
$Realm$	사용자의 공동체
$Times$	시간 • $from$: 요청된 $Ticket$을 사용하길 원하는 시간 표시 • $till$: 요청된 $Ticket$의 사용 만료시간 표시 • $rtime$: 요구되는 재시작부터 만료까지의 시간 표시
$Nonce$	난수 • 응답이 재전송된 것이 아님을 확인
$Subkey$	서브키 • 클라이언트가 선택한 응용 세션을 보호하기 위한 키
$Seq\#$	순서 번호 • 서버가 세션동안 클라이언트에게 보내는 메시지에 재전송을 감지
$Options$	선택사항 • $Ticket$안의 플래그 설정 요청

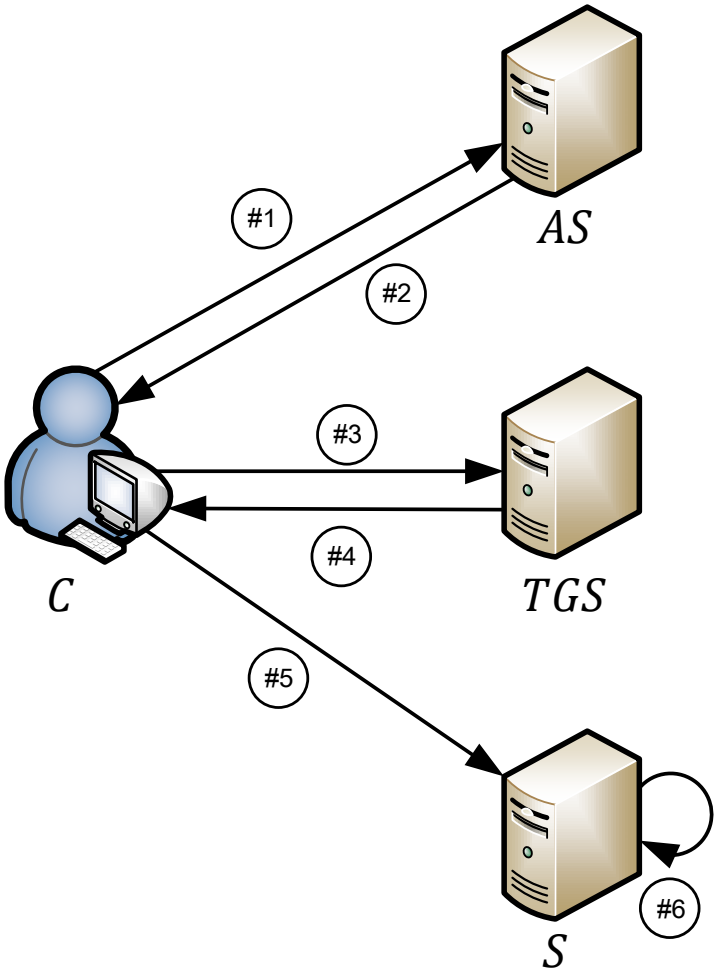
Kerberos

- Kerberos 버전 5
  - $Options$

플래그	설명
INITIAL	AS를 이용하여 발행
PRE-AUTHENT	C는 티켓이 발행되기 전에 KDC에 의해 인증
HW-AUTHENT	하드웨어의 이용이 필요한 초기 인증을 요구
RENEWABLE	교체 티켓을 얻기 위해 재사용될 수 있음
MAY-POSTDATE	날짜가 지난 티켓이 발행
POSTDATED	날짜가 지났음
INVALID	사용 전에 KDC에 의해 유효성 검증이 필요
PROXIABLE	티켓이 대리인임을 나타냄
PROXYFORWARDABLE	다른 네트워크 주소를 이용하여 발행 가능
FORWARDED	인증을 기반으로 발행

Kerberos

- Kerberos 버전 5
  - 동작 과정



인증 서비스 교환: *Ticket*-발행 *Ticket* 취득

#1 $Options \,||\, ID_C \,||\, Realm_C \,||\, ID_{TGS} \,||\, Times \,||\, Nonce_1$ 전송

#2 $Realm_C \,||\, ID_C \,||\, Ticket_{TGS} \,||\, E(K_C, [K_{C-TGS} \,||\, times \,||\, Nonce_1 \,||\, Realm_C \,||\, ID_{TGS}])$ 전송

- $Ticket_{TGS} = E(K_{AS-TGS}, [Flags \,||\, K_{C-TGS} \,||\, Realm_C \,||\, ID_C \,||\, AD_C \,||\, Times])$

***Ticket*-발행 서비스 교환: 서비스-발행 *Ticket* 취득**

#3 $Options \,||\, ID_S \,||\, Times \,||\, Nonce_2 \,||\, Ticket_{TGS} \,||\, Authenticator_{C-TGS}$ 전송

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

#4 $Realm_C \,||\, ID_C \,||\, Ticket_S \,||\, E(K_{C-TGS}, [K_{C-S} \,||\, times \,||\, Nonce_2 \,||\, Realm_S \,||\, ID_S])$ 전송

- $Ticket_S = E(K_{TGS-S}, [Flags \,||\, K_{C-S} \,||\, Realm_C \,||\, ID_C \,||\, AD_C \,||\, Times])$

클라이언트/서버 인증 교환: 서비스 취득

#5 $Options \,||\, Ticket_S \,||\, Authenticator_{C-S}$ 전송

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

#6 상호간의 인증이 필요한 경우, $E(K_{C-S}, [TS_2 \,||\, Subkey \,||\, Seq\#])$ 전송

Kerberos

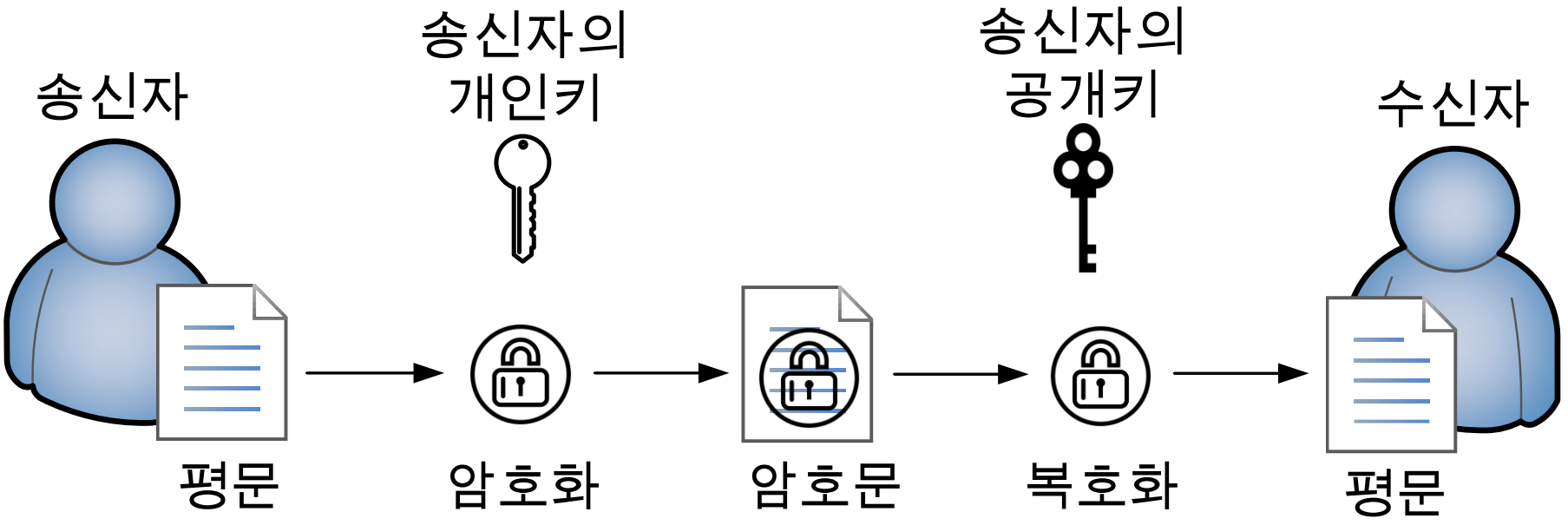
- Kerberos 버전 5
  - 장점
    - 사용자와 서비스 간의 통신 내용을 암호화 키를 이용하여 보호하기 때문에 데이터의 기밀성 및 무결성을 보장
  - 단점
    - 모든 사용자와 서비스의 암호화 키를 KDC가 가지고 있어 KDC에 오류 발생시 전체 서비스 사용 불가
    - 사용자의 패스워드가 유출되어도 서버는 이를 인식하지 못함
    - 사용자가 패스워드를 바꾸면 비밀키도 변경해야 함

목 차

- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

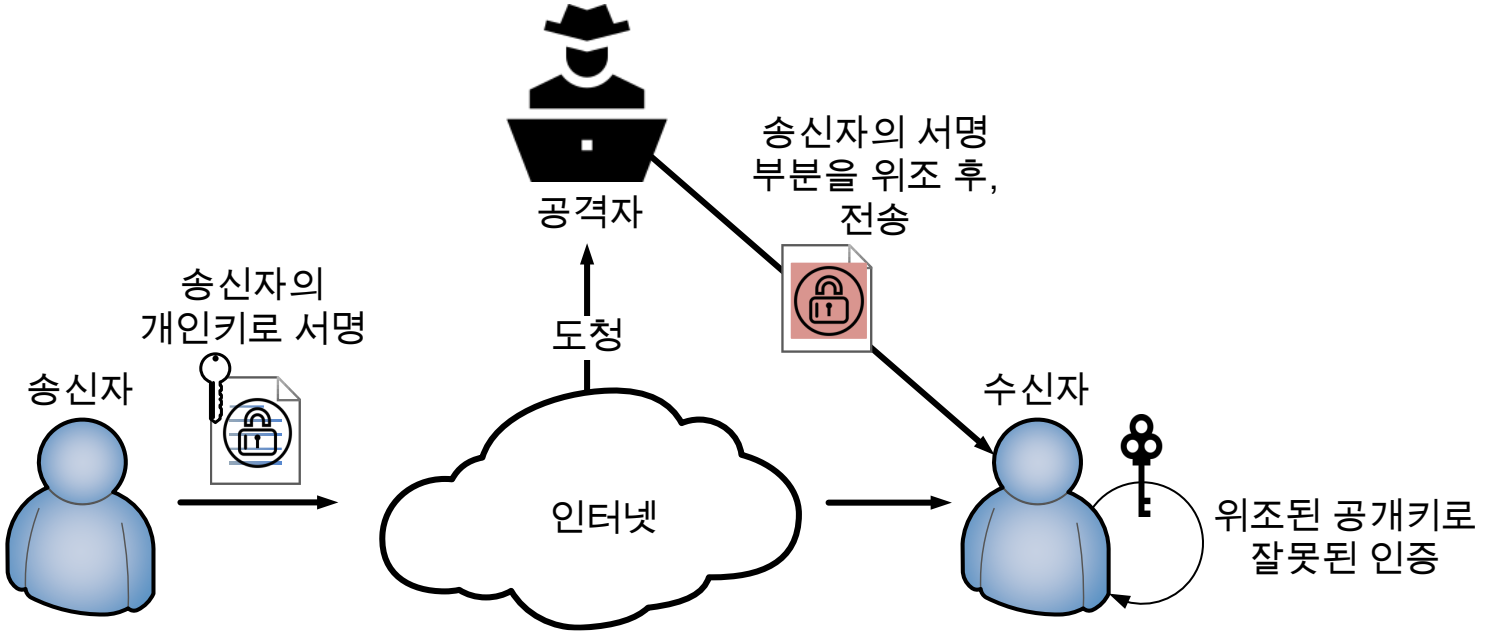
비대칭 암호의 키 분배

- 공개키 암호
  - 정의
    - 외부에 공개되는 공개키와 자기 자신만 알 수 있는 개인키의 사용으로 암호화하는 기법



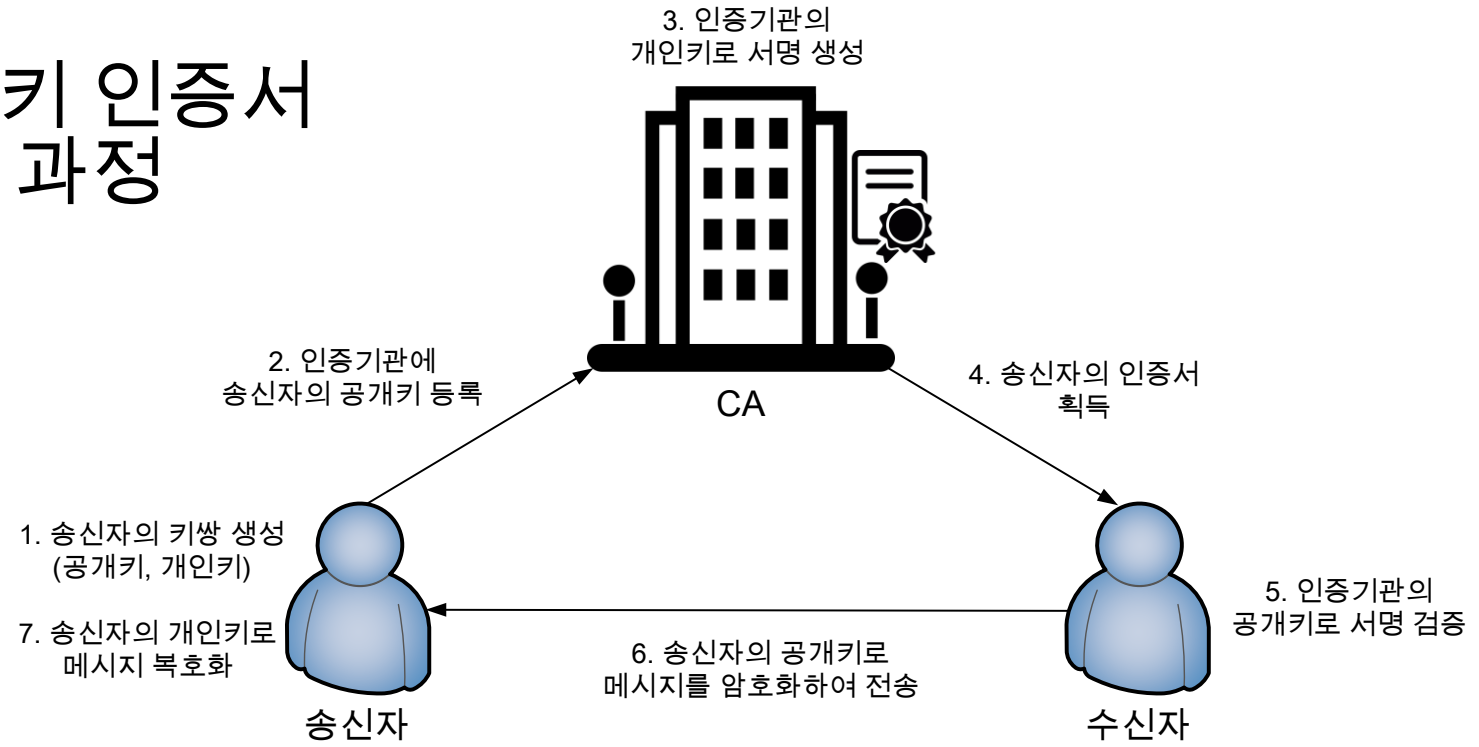
비대칭 암호의 키 분배

- 공개키 암호
  - 위협
    - 통신 간의 메시지를 도청하여 메시지 서명 부분을 공격자의 서명으로 위조
    - 공격자의 공개키를 수신자에게 전송함으로써, 공격자는 신분위장 가능



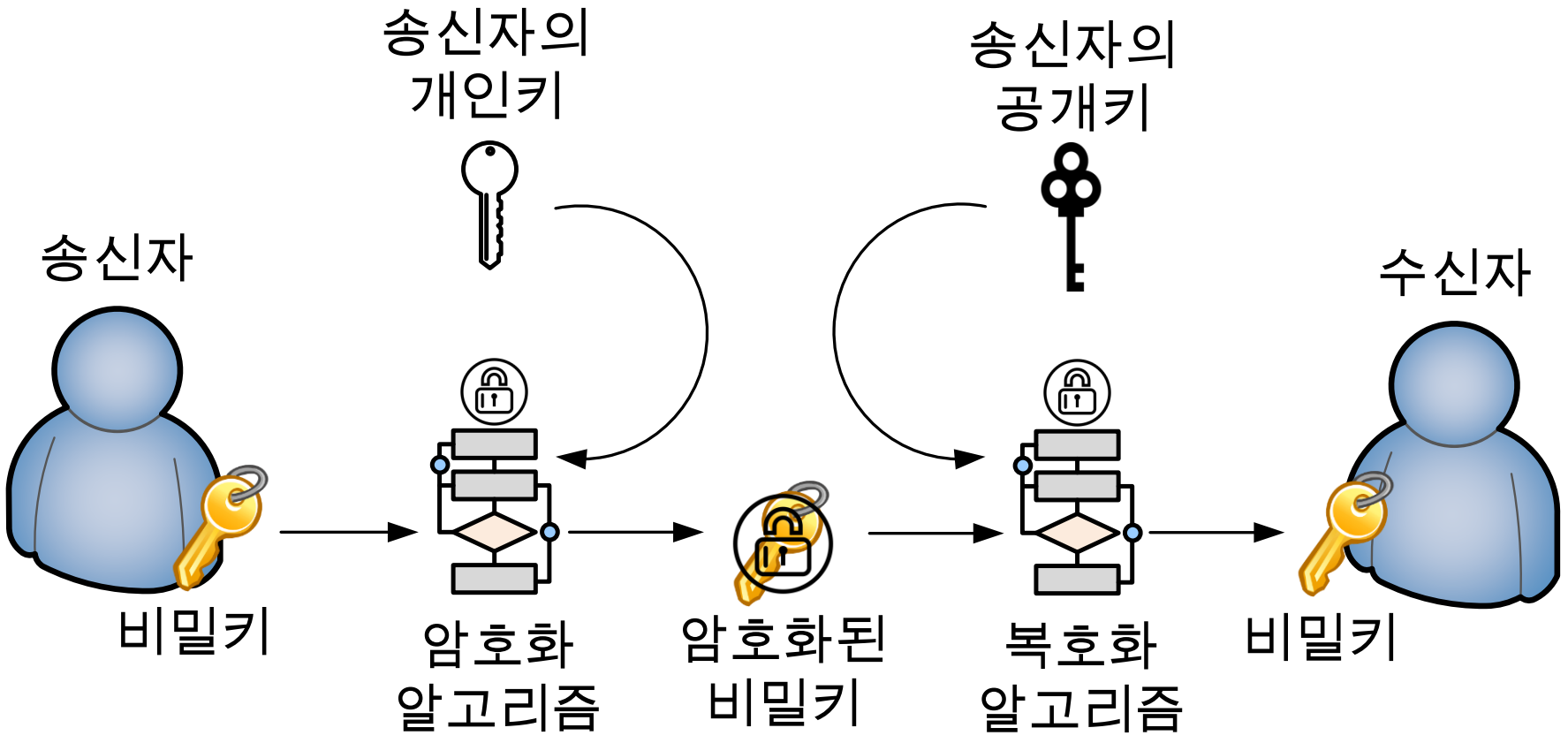
비대칭 암호의 키 분배

- 공개키 인증서
  - 정의
    - 공개키와 사용자ID로 구성되어 있는 메시지를 신뢰할 수 있는 제 3자가 서명한 것
      - e.g., 인증 기관(CA, Certificate Authority), 금융 기관 등
  - 공개키 인증서 동작 과정



비대칭 암호의 키 분배

- 키 분배 과정
  - 양측이 통신하는데, 안전하게 비밀키를 공유하기 위해 사용



목 차

- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

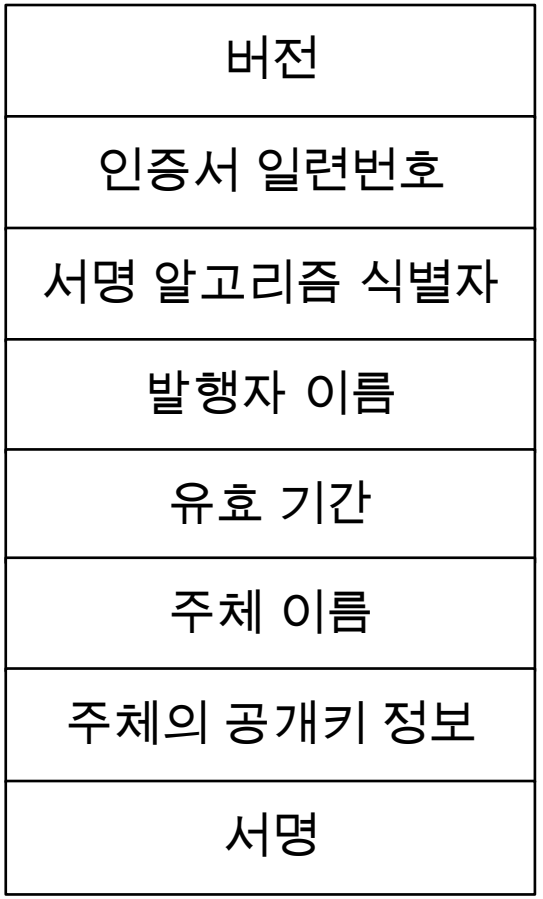
X.509 인증서

- 정의
  - 공개키 암호 기반구조의 인증 서비스를 구현하기 위한 표준 인증서 형식
- 특징
  - 공개키 인증서를 이용한 인증 프로토콜
  - 디지털 서명 사용으로 부인방지 제공
  - 위조 불가능
  - S/MIME, IP Security, SSL/TLS 등에 사용

X.509 인증서

- 버전 1 형식

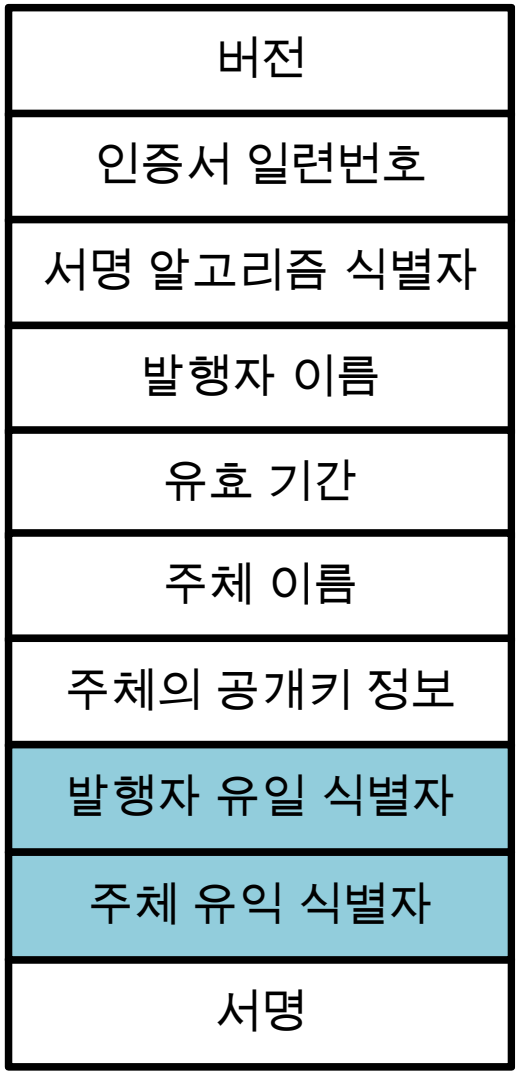
항목	설명
버전	인증서 형식을 구별하기 위한 번호
일련번호	인증기관에서 발행한 인증서의 일련번호
서명 알고리즘 식별자	인증서 서명에 사용한 알고리즘
발행자 이름	인증서를 만들고 서명한 인증기관의 이름
유효기간	인증서가 유효한 시작일과 만료일
주체이름	인증서가 인증하는 사용자의 이름
주체의 공개키 정보	주체의 공개키와 이 키가 사용하는 알고리즘 식별자
서명	인증기관의 개인키로 암호화한 해시값



X.509 인증서

- 버전 2 형식
  - 발행자 유일 식별자 추가
  - 주체 유일 식별자 추가

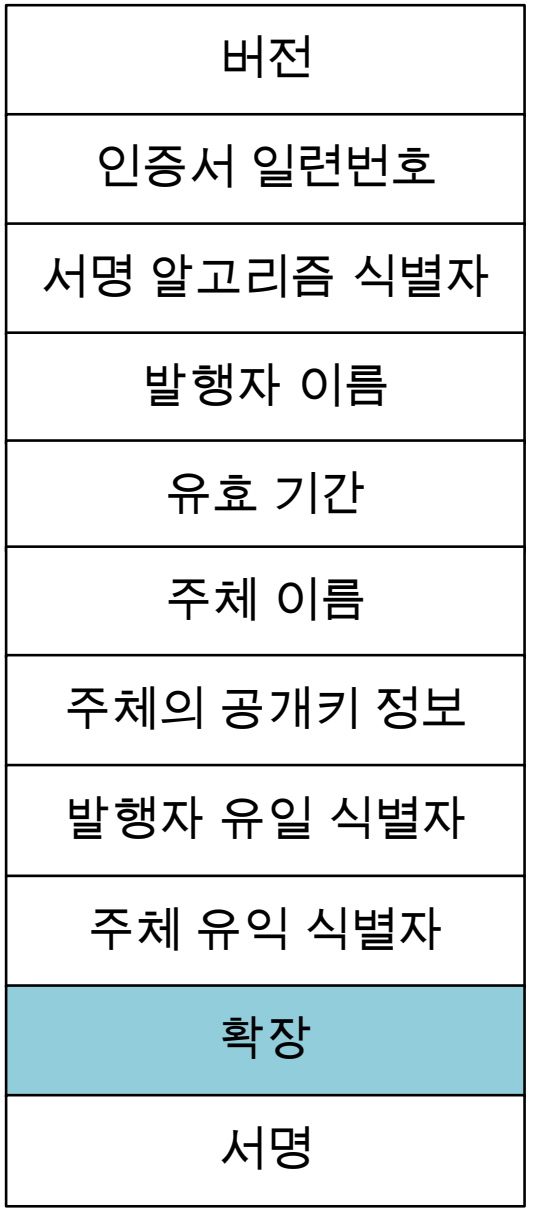
항목	설명
발행자 유일 식별자	인증기관을 식별하기 위한 고유번호
주체 유일 식별자	인증서 주체를 식별하기 위한 고유 번호



X.509 인증서

- 버전 3 형식
  - 확장 추가(1/3)
    - 키와 정책 정보

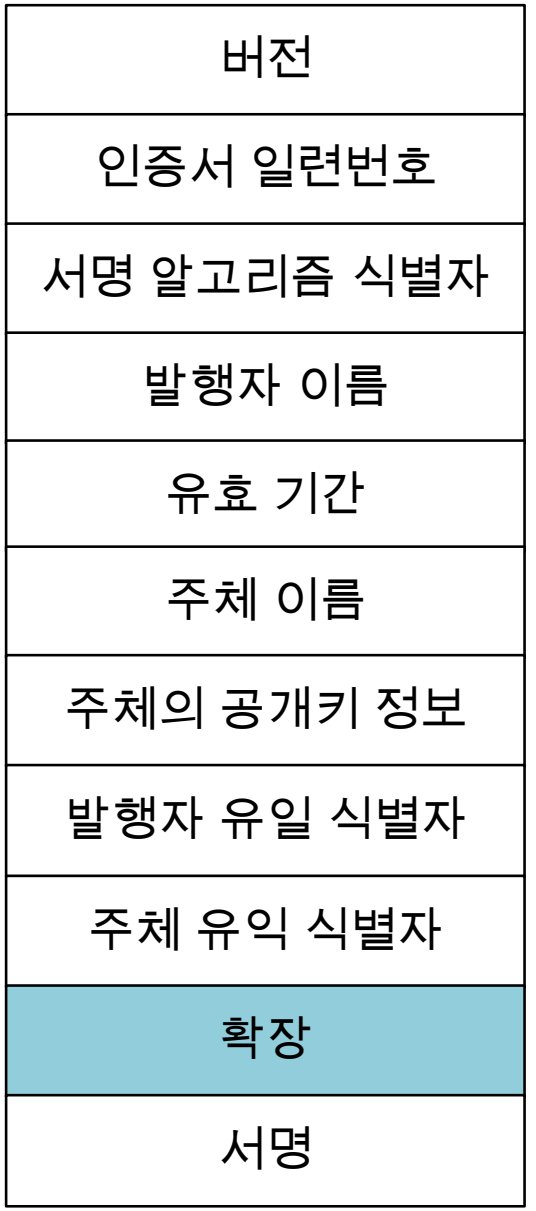
키와 정책 정보	설명
기관 키 식별자	인증서나 CRL에 있는 서명을 확인하는데 쓰이는 공개키를 식별
주체 키 식별자	인증될 공개키를 식별
키 용도	어떤 목적인지 어떤 정책하에 사용될 것인지에 따라 키에 대한 제한조건
개인키 유효기간	공개키와 대응되는 개인키의 사용기간
인증서 정책	해당 인증서가 지원되는 것으로 인식하는 정책 항목
정책 매핑	• 다른 CA에 의해 발행한 인증서에만 사용됨 • 인증서를 발행한 CA의 정책이 발행 받은 다른 CA에 적용되는 다른 정책과 동일하게 간주될 수 있도록 함



X.509 인증서

- 버전 3 형식
  - 확장 추가(2/3)
    - 인증서 주체와 발행자 속성

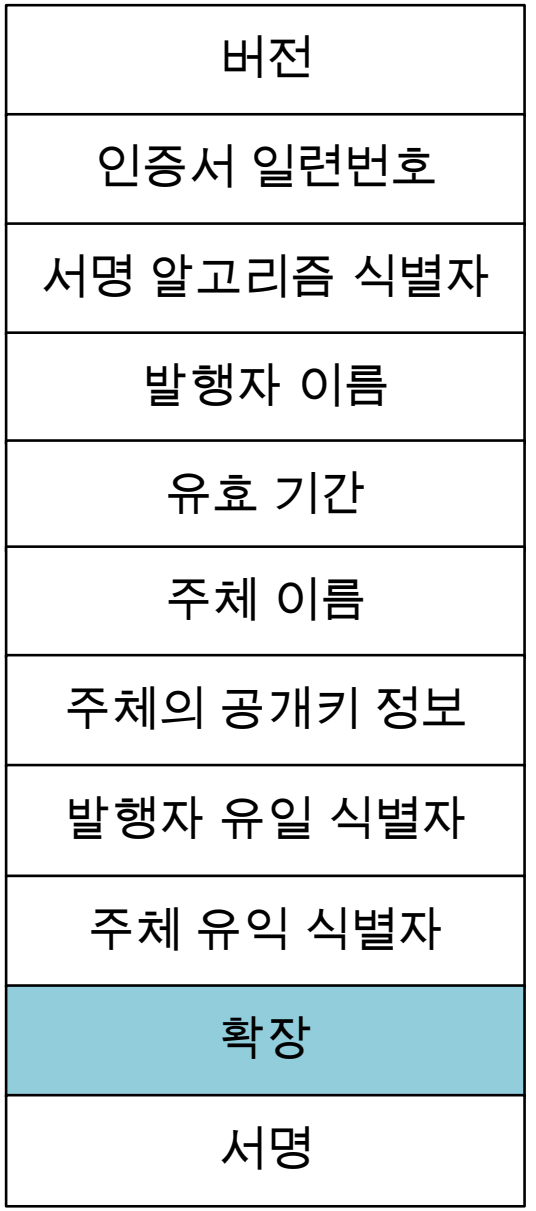
인증서 주체와 발행자 속성	설명
주체 대체 이름	• 자체적으로 이름 형식을 가지고 있는 특정 응용프로그램을 지원하기 위한 대체이름 • e.g., 전자메일, IPsec
발행자 대체 이름	• 발행자의 대체 이름
주체 디렉터리 속성	• 인증서의 주체를 위한 X.500 디렉터리 속성값



X.509 인증서

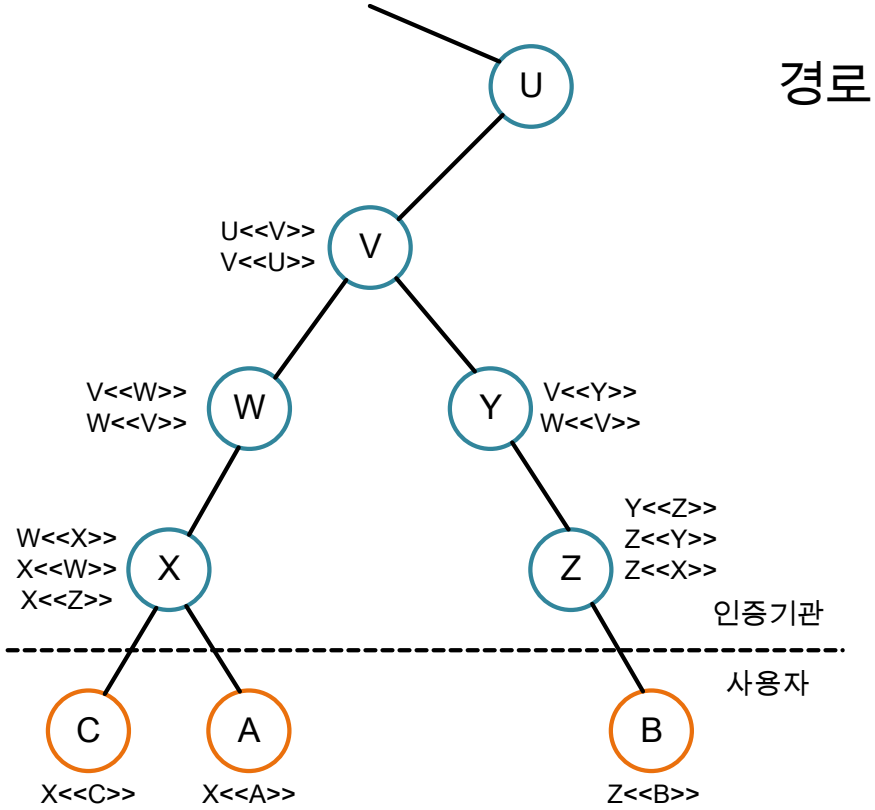
- 버전 3 형식
  - 확장 추가(3/3)
    - 인증경로 제약조건

인증경로 제약조건	설명
기본 제약	• 주체가 CA 역할을 하는지에 대한 여부 • 인증 경로에 대한 정보
이름 제약	• 인증서에 나타나는 모든 주체의 이름이 위치해야 하는 이름 공간
정책 제약	• 구체적인 인증서 정책 식별 요구 • 인증서 정책 매핑 금지 요구

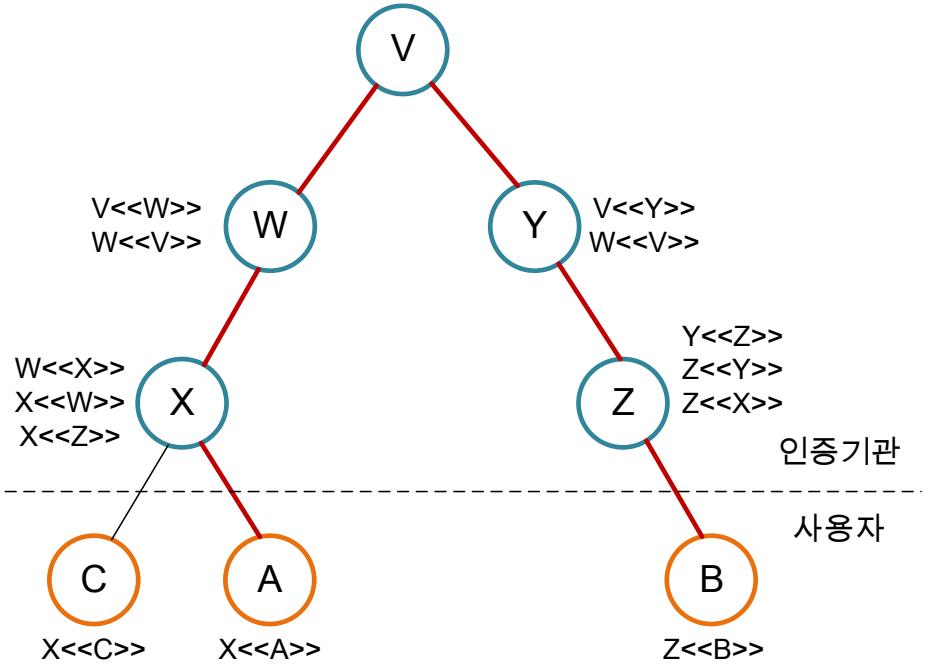


X.509 인증서

- 인증서 체인(Chain of Certificate)
  - 서로 다른 기관으로부터 안전하게 공개키를 획득하기 위해서 인증 기관들을 계층구조(트리구조)로 연결한 것
    - 사용자 A에서 사용자 B의 인증서를 얻기 위한 인증 경로

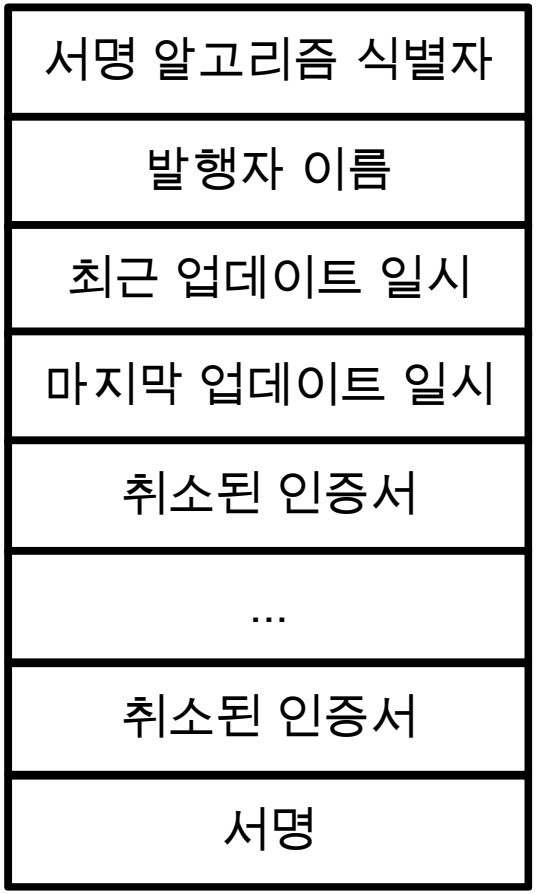


경로: A<<X>>X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>



X.509 인증서

- 인증서 취소
  - 인증서 취소 목록
    (CRL, Certificate Revocation List)
  - 인증서 유효기간 만료 이외의 취소 사유
    - 사용자 개인키가 노출되거나 훼손된 경우
    - CA가 사용자를 더 이상 인증해 줄 수 없는 경우
    - CA의 인증서가 노출되거나 훼손된 경우



목 차

- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

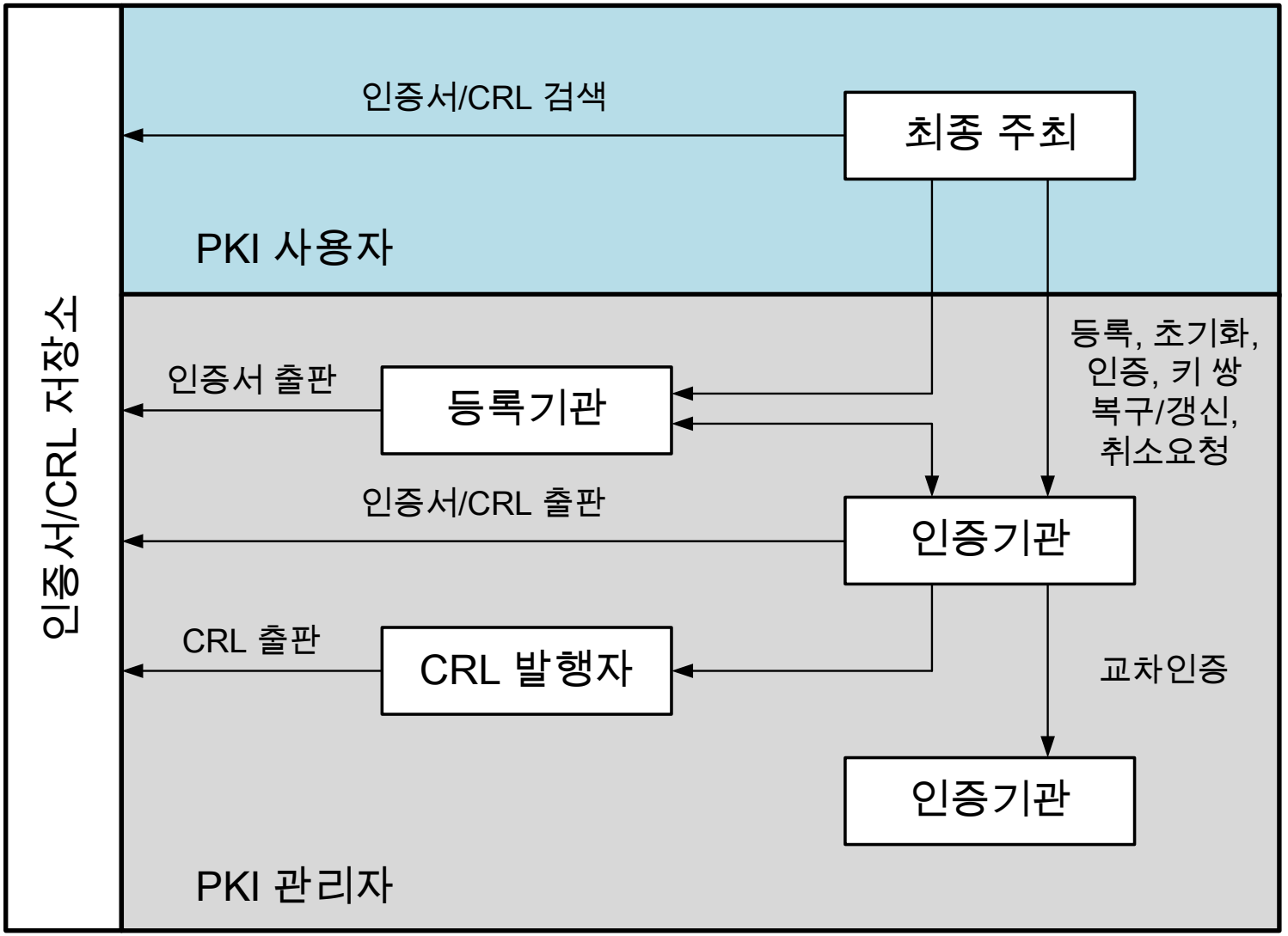
공개키 기반 구조

- PKIX(Public-key Infrastructure X.509)
  - X.509에 기초하여 디지털 인증서를 생성, 관리, 저장, 분배, 취소하는데 필요한 하드웨어, 소프트웨어, 사용자, 정책 및 절차
  - 구성 요소

구성요소	설명
사용자	종단 사용자, 종단 장치 등으로 PKI와 관련된 서비스를 사용
인증기관(CA)	인증서와 인증서 취소 목록(CRL)을 발행
등록기관 (RA, Registration authority)	선택적 요소로 사용자의 등록절차와 관련된 업무 수행
CRL 발행자 (CRL issuer)	선택적 요소로 인증서 취소 목록 발행
저장소 (Repository)	인증서와 인증서 취소 목록을 저장하며 사용자가 검색할 수 있는 장소

공개키 기반 구조

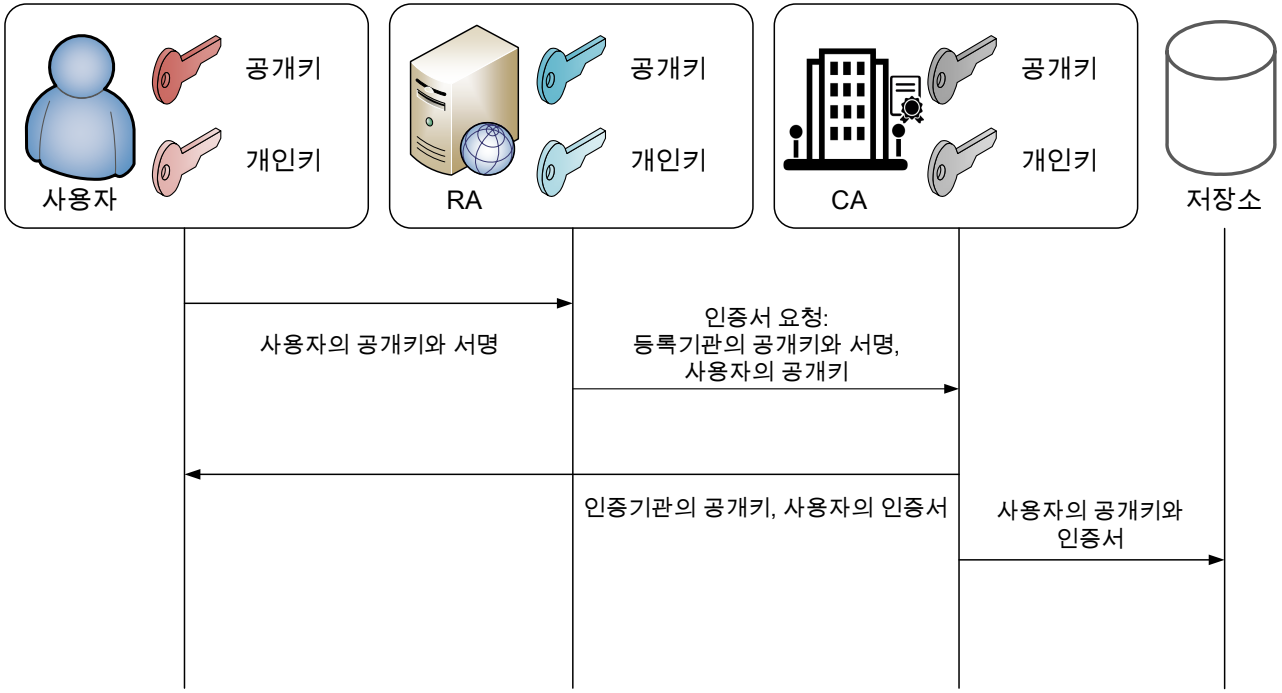
- PKIX 구조 모델



공개키 기반 구조

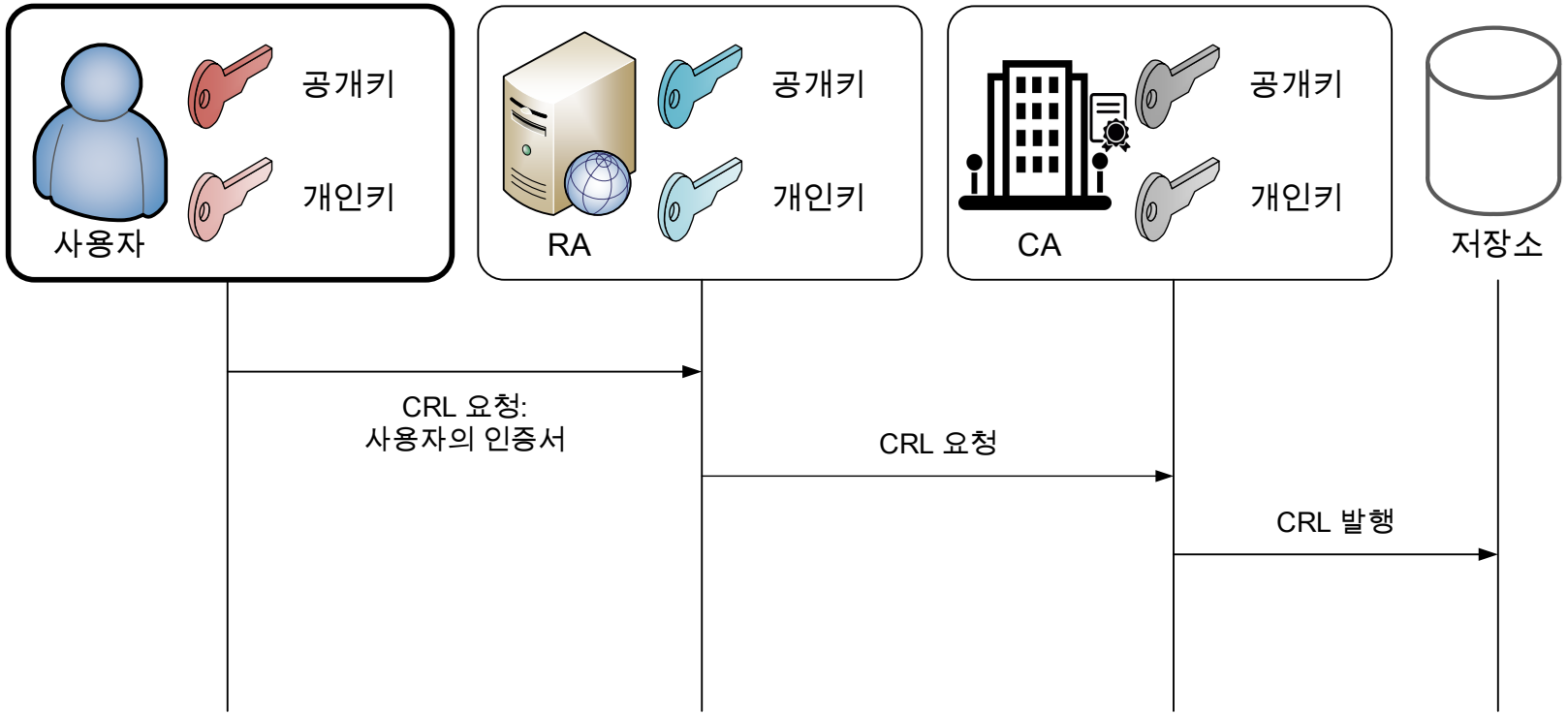
- PKIX 관리 기능

관리기능	설명
등록(Registration)	사용자가 최초로 CA에게 자신을 알리는 절차
초기화(Initialization)	사용자의 시스템이 안전하게 작동되기 전에 키와 관련된 요소들을 설치 (e.g., 사용자의 공개키, 신뢰받는 CA 정보)
인증(Certification)	사용자에게 인증기관의 공개키를 발급하고 인증서를 사용자의 시스템에게 인증서를 돌려주며 인증서를 저장소에 저장



공개키 기반 구조

- PKIX 관리 기능
  - 취소요청(Revocation Request)
    - 정상적인 키 사용이 불가능한 경우, 인증서를 폐지하도록 요청
      - e.g., 개인키 분실, 도난, 이름 변경 등



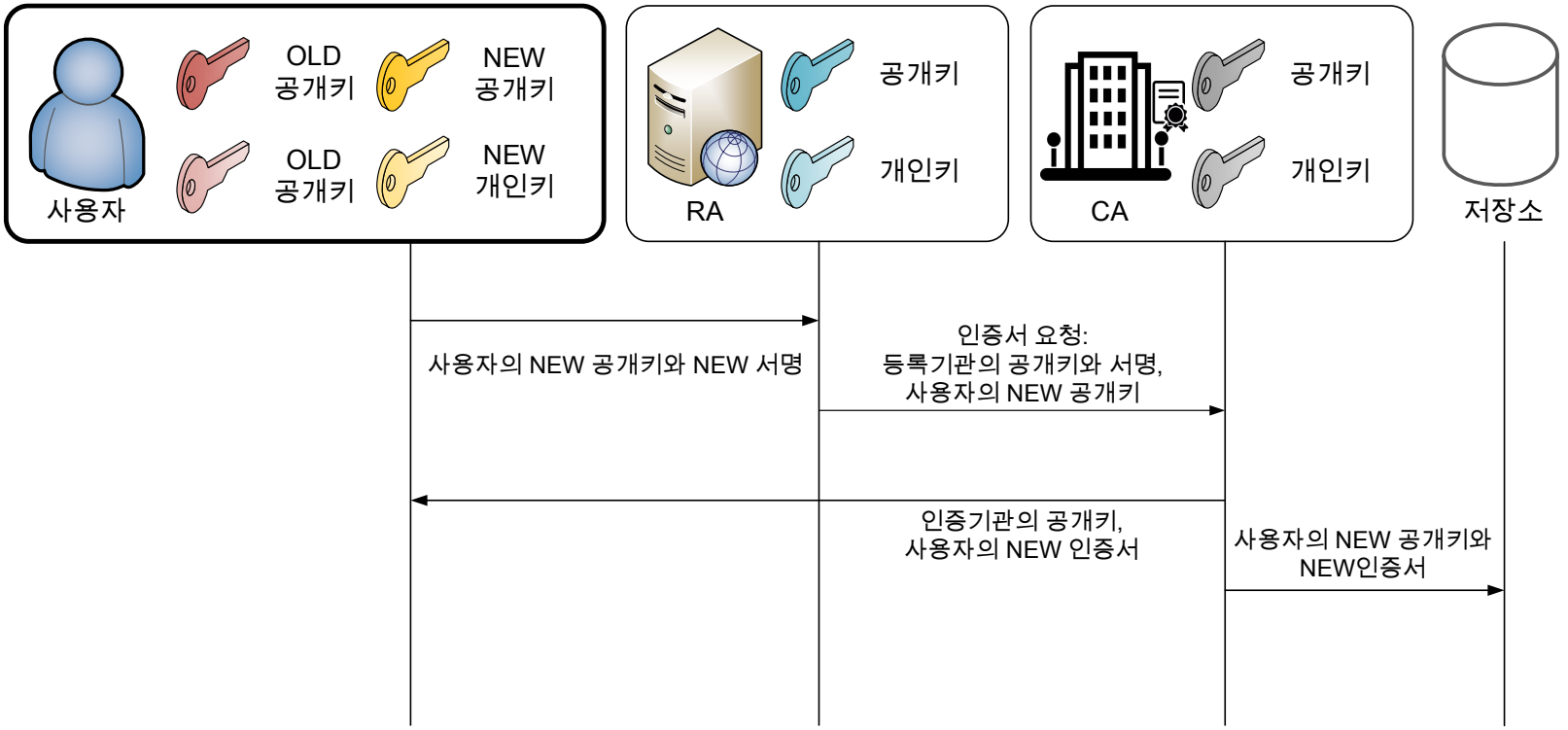
공개키 기반 구조

- PKIX 관리 기능
  - 키 쌍 복구(Key Pair Recovery)
    - 정상적인 키 사용이 불가능한 경우를 대비하여 개인키를 획득하는 절차

방식	설명
위탁 방식	• 사용자의 비밀키 전부 또는 일부를 신뢰받는 제 3자에게 위탁하는 방식 • 신뢰도가 낮은 위탁기관을 사용하는 경우, 문제발생
캡슐화 방식	• 각각의 메시지 전송 마다 키 복구에 필요한 정보를 담은 영역을 생성 후, 해당 메시지를 복구할 수 있는 정보를 데이터에 부가하는 방식 • 복구 영역을 사용자가 생성하기 때문에 수정과 조작이 용이 • 프로토콜 자체가 복구 영역에 대한 확장 영역을 지원해야 함
TTP(Trusted Third Party) 방식	• 신뢰할 수 있는 제 3자인 TTP를 가정하여, 복구될 사용자의 비밀키를 사용자의 신뢰기관에서 생성하고 사용자에게 분배하는 방식 • 많은 개인의 정보가 TTP에게 의존

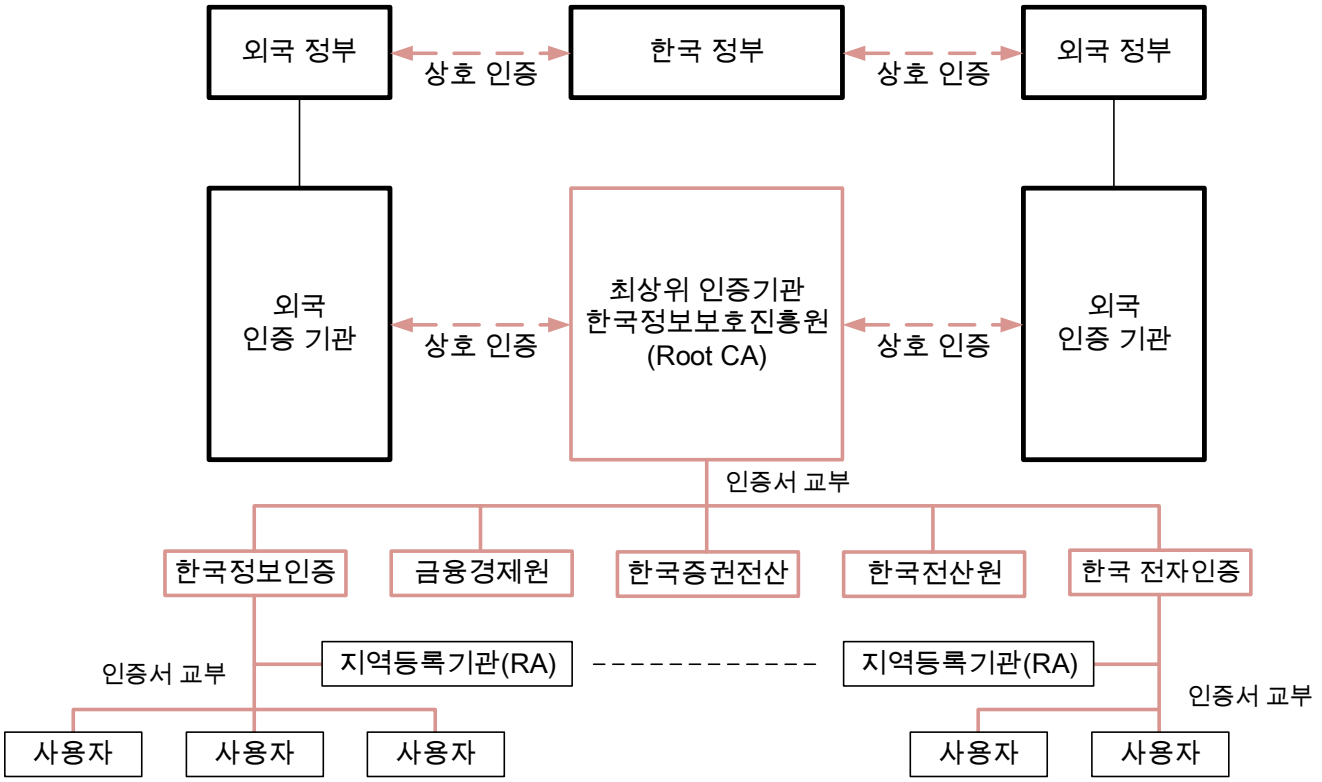
공개키 기반 구조

- PKIX 관리 기능
  - 키 쌍 갱신(Key Pair Update)
    - 모든 키 쌍을 정기적으로 갱신하기 위해서 새로운 인증서를 발급
      - e.g., 인증서의 유효기간 종료된 경우, 인증서가 취소된 경우



공개키 기반 구조

- PKIX 관리 기능
  - 교차인증(Cross Certification)
    - 한 CA가 다른 CA에게 인증서를 발급하며 두개의 CA가 정보를 교환하여 서로 다른 CA의 인증서를 가진 사용자를 인증



목 차

- 대칭키 암호의 키 분배
- Kerberos
- 비대칭 암호의 키 분배
- X.509 인증서
- 공개키 기반 구조
- 통합신원관리

통합신원관리

- 신원관리(Identity Management)
  - 정의
    - 중앙 집중화 방식으로 사용자가 시스템에 접근하는 것을 통제하는 절차
    - 사용자의 신원을 정의하고 신원과 연관된 속성으로 사용자 신원을 인증
- 싱글사인온(SSO, Single Sign-on)
  - 정의
    - 사용자가 한번만 인증하면, 네트워크의 모든 자원에 접속할 수 있게 하는 방식

통합신원관리

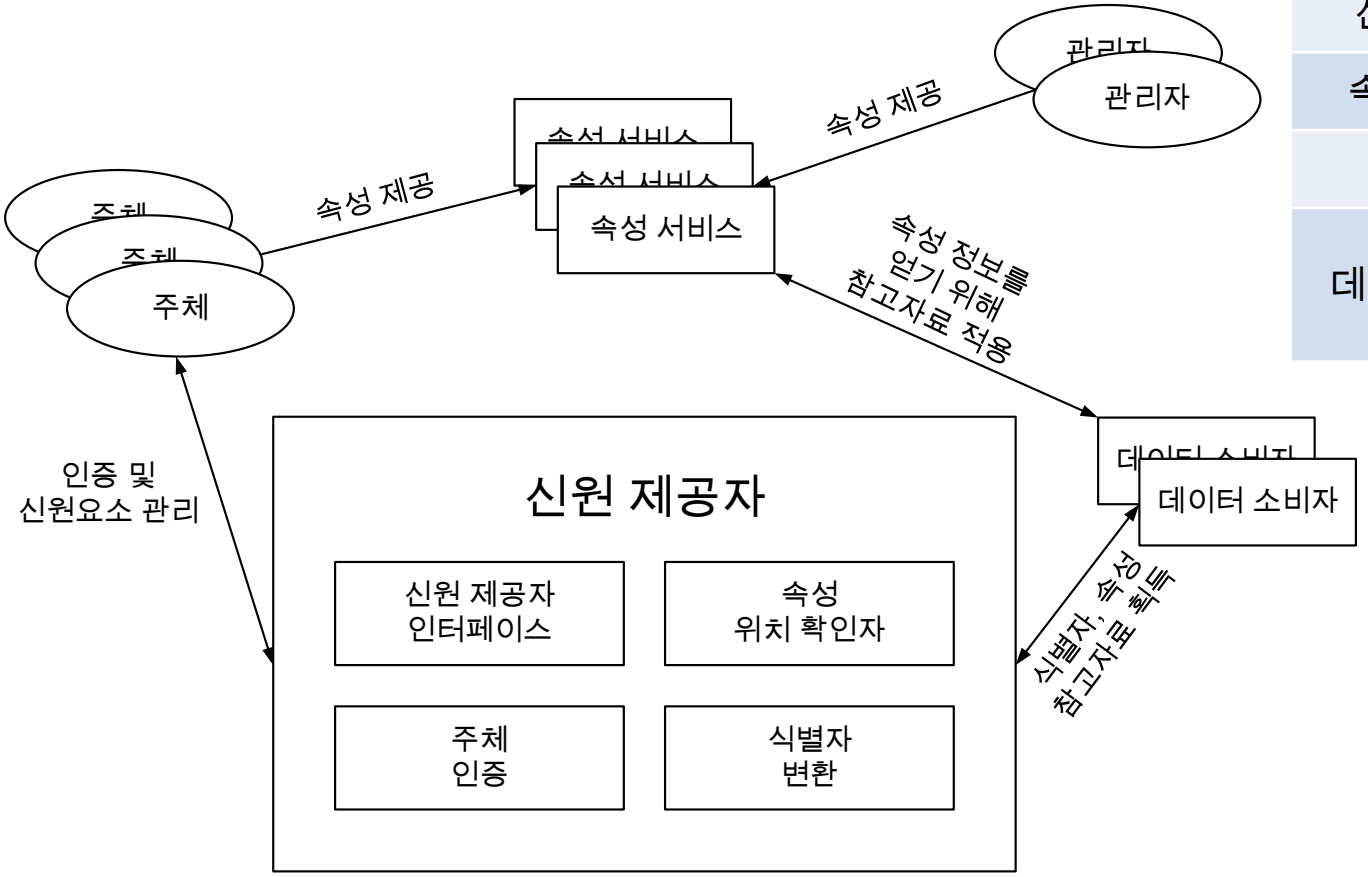
- 신원관리(Identity Management)
  - 원칙

원칙	설명
인증(Authentication)	제공한 사용자 이름과 사용자가 일치하는지 확인하는 것
허가(Authorization)	인증에 근거해서 특정 서비스나 자원에 접근을 허락하는 것
계정(Accounting)	로그인 접근과 허가 절차
제공(Provisioning)	시스템에 사용자 등록
작업절차 자동화 (Workflow automation)	비즈니스 프로세스에서 데이터의 이동
관리위임 (Delegated administration)	허가부여를 위한 역할-기반 접근 통제 사용
패스워드 동기화 (Password synchronization)	SSO를 생성하여 사용자가 일회의 인증으로 네트워크의 모든 자원에 접근할 수 있도록 함
셀프-서브 패스워드 리셋 (Self-service password reset)	사용자가 자신의 패스워드를 갱신하도록 함
통합 (Federation)	인증과 허가절차를 한 시스템에서 다른 시스템으로 전달하여 사용자가 매번 인증 받을 필요가 없음

통합신원관리

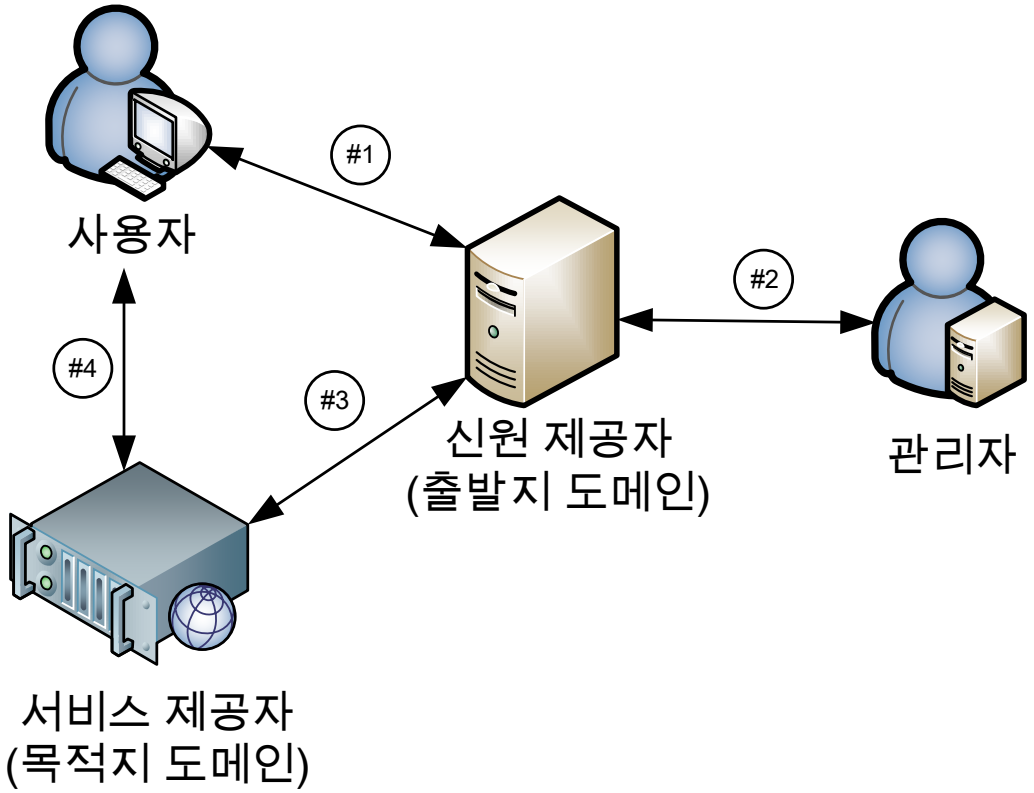
- 신원관리(Identity Management)
  - 일반적인 구조

용어	설명
주체	신원 소비자
신원제공자	인증정보를 주체와 교류
속성 서비스	속성정보 생성/유지/관리
관리자	사용자에게 속성을 부여
데이터 소비자	신원 제공자 또는 속성 서비스가 관리하는 데이터를 받아 사용하는 개체



통합신원관리

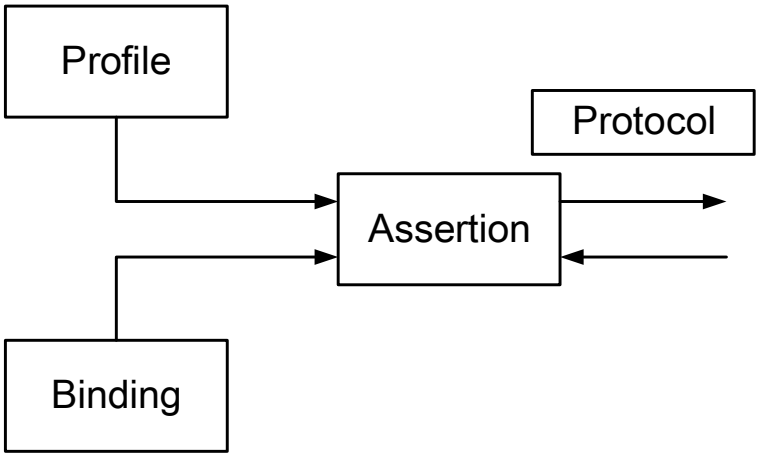
- 신원통합(Identity Federation)
  - 다른 도메인으로 신원관리를 확장하고 디지털 신원을 공유
    - 한번의 사용자 인증으로 다수의 도메인에 있는 응용 프로그램이나 자원에 접근할 수 있도록 하는 것



순서	설명
#1	동일 도메인 내의 신원 제공자에게 사용자는 신원과 연관된 속성 값을 제공
#2	동일 도메인 내의 관리자가 제공하는 신원과 이와 연관된 속성 값 제공
#3	원격 도메인 서비스 제공자에게 사용자의 신원 정보와 인증정보 제공
#4	서비스 제공자는 원격 사용자와 세션을 생성하고 사용자의 신원 및 속성에 기반한 제한요건에 맞춰 접근 통제

통합신원관리

- 신원통합 표준
  - SAML(Security Assertion Markup Language)
    - 보안 도메인 간에 인증정보와 권한부여에 관련된 자료를 교환할 수 있도록 설계된 표준
  - SAML 구조

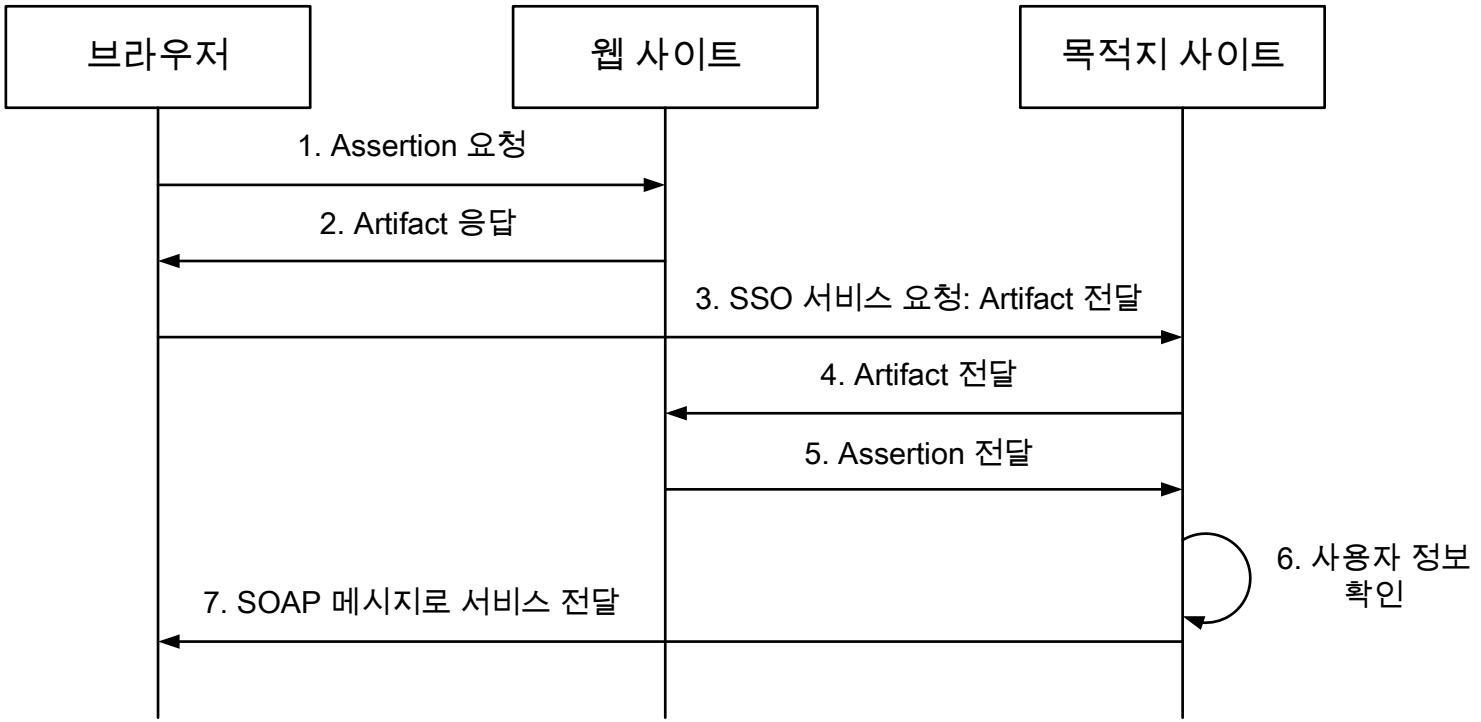


용어	설명
Assertion	사용자가 원하는 웹사이트에 대한 접근하기 위한 정보
Profile	메시지 안의 사용자 정보
Binding	요청과 응답을 전송하는 방식
Protocol	요청과 응답의 포맷 결정

통합신원관리

- 신원통합 표준
  - SSO 서비스 시나리오

용어	설명
Artifact	시스템이에서 자동으로 생성한 메시지
SOAP 메시지	애플리케이션 간에 정보를 교환하기 위한 메시지



Thanks!

박 재 형 (jaehyoung@pel.sejong.ac.kr)