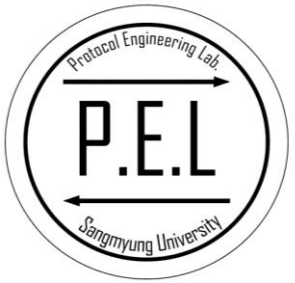


NETWORK SECURITY ESSENTIALS

대칭 암호를 이용한 대칭키 분배 / KERBEROS

Boo-Hyung Lee
(boohyung@pel.smuc.ac.kr)
Protocol **E**ngineering **L**ab. **Sangmyung** University

2015. 01. 06

Content

- 대칭 암호를 이용한 대칭 키 분배

- KERBEROS

대칭 암호를 이용한 대칭 키 분배

- **대칭키 암호 방식**

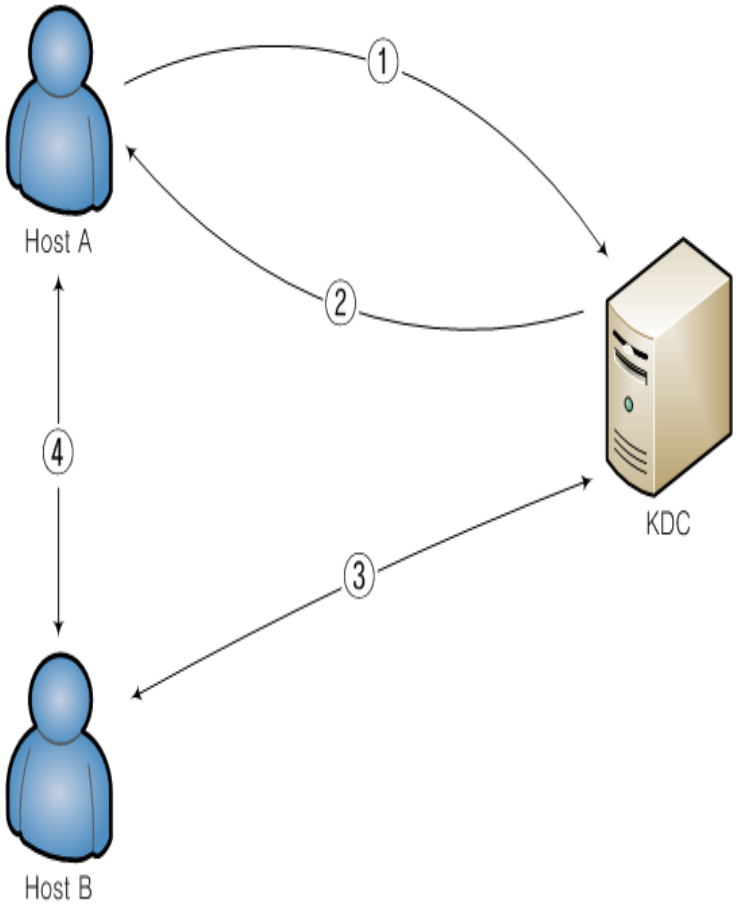
  - 수신자와 송신자가 반드시 동일한 키를 공유하고 있어야 가능
  - 사전에 당사자끼리 키를 공유하는 과정이 필요

- **이처럼 중요한 키를 어떻게 분배해야 할까?**

  - 다른 사람이 키를 보지 못하게 하여 **"암호화"**
  - 데이터를 교환하고자 하는 쌍방에게 전달해야 함 **"안전한 수단을 이용"**

대칭 암호를 이용한 대칭 키 분배

- **KDC를 사용한 대칭 키 분배**



① B와의 통신을 요청

- A와 KDC가 공유하는 비밀키로 암호화하여 전송

② 요청에 대한 응답 : 세션 키 전달

- A와 KDC가 공유하는 비밀키로 암호화하여 전송

③ A의 요청을 전달하고 B가 동의하면 세션 키 전달

- B와 KDC가 공유하는 비밀키로 암호화하여 전송

④ 통신

- 메시지는 KDC로부터 전달받은 세션 키로 암호화

☞ KDC(Key Distribution Center, 키 분배 센터)

- 사전에 클라이언트들과 공유하고 있는 비밀 키를 보관하고 관리

☞ 세션 키

- 클라이언트들간의 통신내용을 암호화하는데 사용하는 비밀키
- 세션이 유지되는 동안 사용하고, 세션이 종료되면 폐기

대칭 암호를 이용한 대칭 키 분배

- **KDC를 이용할 때의 문제점 : 키의 보관**

- 클라이언트가 늘어나면 KDC에서 보관해야 할 키의 숫자가 늘어남
- 만약, 클라이언트가 n명이라면 n(n+1)/2개의 키를 보관해야 함
- KDC가 공격 당하면 모든 사용자의 키가 노출될 수 있음

예 1> Client A, Client B, KDC (클라이언트의 수 : 2)

- KDC가 보관해야 할 키 : $K_{A\text{-}KDC}$, $K_{B\text{-}KDC}$, $K_{A\text{-}B}$ (총 3개)

예 2> Client A, Client B, Client C, KDC (클라이언트의 수 : 3)

- KDC가 보관해야 할 키 : $K_{A\text{-}KDC}$, $K_{B\text{-}KDC}$, $K_{C\text{-}KDC}$, $K_{A\text{-}B}$, $K_{A\text{-}C}$, $K_{B\text{-}C}$ (총 6개)

KERBEROS

- **KERBEROS 개요**

- MIT에서 개발한 비밀키 암호 기반 키 분배 및 사용자 인증 시스템
- version 4는 1980년대 말에 처음 Project Athena를 통해 처음 알려졌으며, version 5는 RFC 1510(1993), RFC 4120(2005)를 통해 발표되었음
- 중앙 집중식 인증 서버 이용
- 개방된 네트워크 내에서 사용하는 서비스 요구 인증 방법
- UDP 88번 포트를 사용; 티켓 사이즈가 커지는 경우에는 TCP 88번 포트사용

참고) IP Packet의 max size는 65535, IP header size 20, UDP header size 8

→ Max UDP Payload size = 65507 bytes

- 현재 Kerberos ver.5 까지 출시 되었음(windows용은 ver.4까지 출시됨)

http://web.mit.edu/kerberos/dist/index.html#kfw-4.0

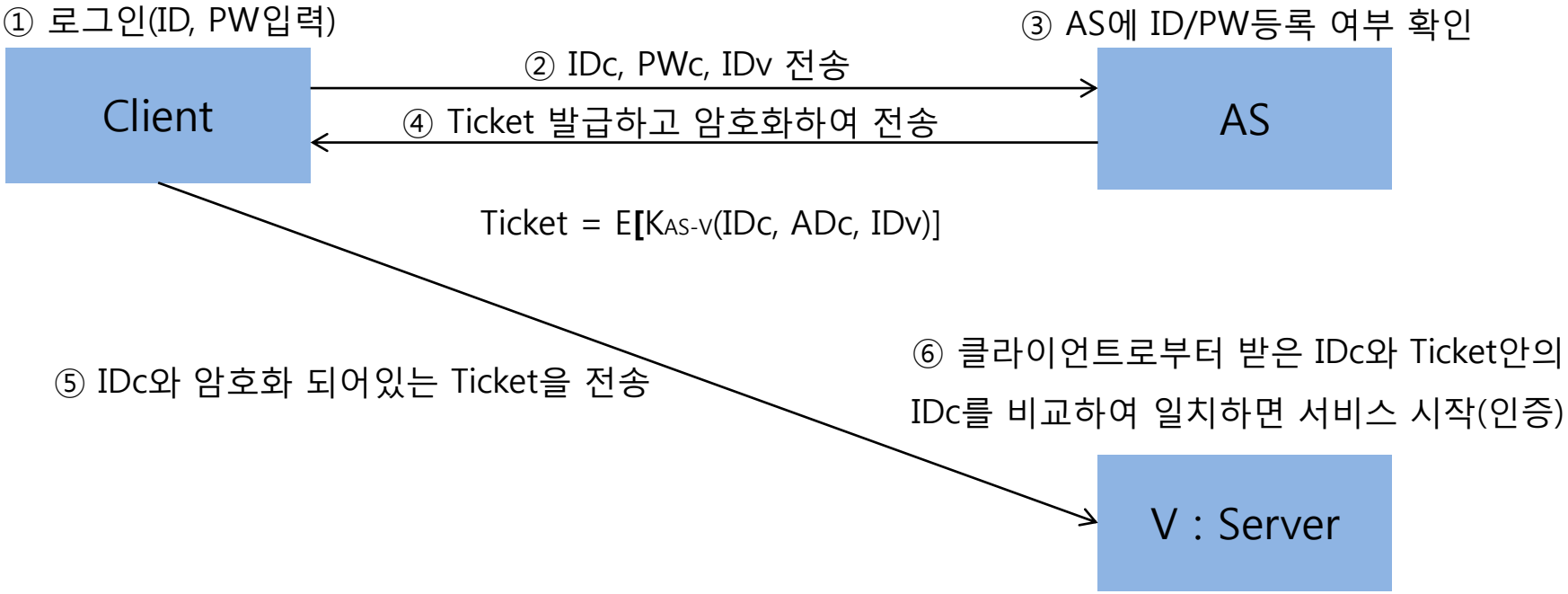
KERBEROS

- **구성요소**

- **클라이언트 :** 사용자의 컴퓨터
- **서버** : 사용자가 접속하려고 하는 대상; 서버에 접속하려면 인증이 필요
- **인증 서버(AS, Authentication Server) :** 클라이언트를 인증하는 서버; 모든 사용자의 아이디와 패스워드를 데이터베이스에 저장하고 있음; TGT 발급
- **티켓 발행 서버(TGS, Ticket Granted Server) :** 인증 값인 티켓을 클라이언트에게 발급해주는 서버
  → 인증 서버와 티켓 발행 서버를 통틀어 키 분배 센터라 부름(KDC)
- **티켓 승인 티켓(TGT, Ticket Granted Ticket) :** 티켓을 발급받는데 필요한 티켓
- **서비스 승인 티켓(Ticket) :** 클라이언트가 서버에 접속할 때 필요
- **세션 키 :** 서비스 세션 당 한번씩 사용하는 비밀 키
- **인증자(Authenticator) :** 서비스 요청자가 클라이언트 자신임을 인증; 일회용이고 유효기간이 짧기 때문에 위협 가능성이 상대적으로 적음
- **타임스탬프(TS), 유효기간(Lifetime) :** 티켓의 재사용 방지

KERBEROS

- **간단한 인증 프로토콜**



☞ IDc : 클라이언트의 ID, PWc : 클라이언트의 패스워드, IDv : 서버의 ID, ADc : 클라이언트의 IP주소

K_{AS-V} : AS와 Sever가 사전에 공유하고 있는 비밀 키

KERBEROS

- **간단한 인증 프로토콜의 문제점**

- ②번 과정에서 패스워드가 평문으로 전송됨을 확인 : 전송 중간에 패스워드 가로채기 가능

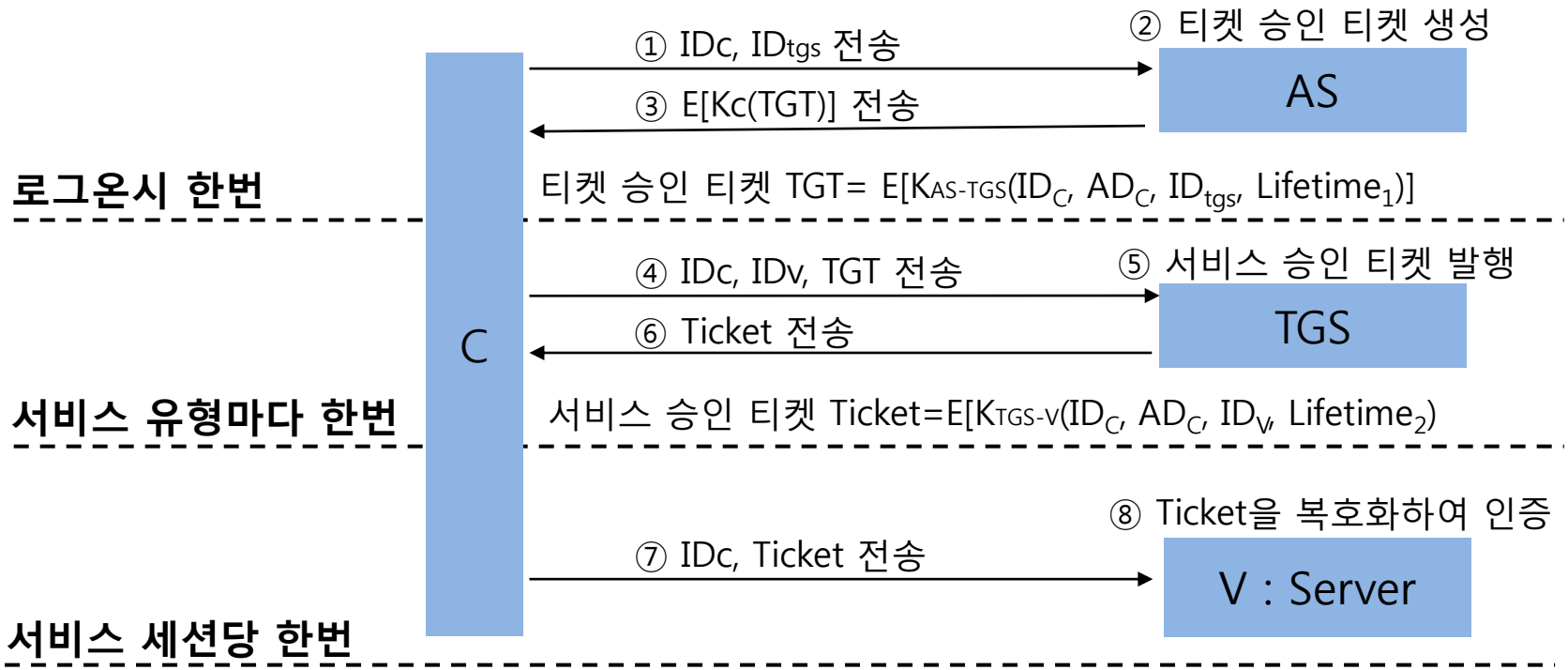
! 패스워드를 암호화하여 전송

- 서비스마다 새로운 티켓이 요구됨
- 다른 서비스 사용에는 패스워드 입력을 다시 해야 하는 불편함

! 티켓 발행 서버(TGS)를 사용하여 해결

KERBEROS

- **TGS를 사용한 인증 프로토콜**



☞ Kc : 패스워드로부터 얻은 키, K_{AS-TGS} : AS와 TGS가 사전에 공유하고 있는 비밀 키, ID_{tgs} : TGS의 ID
K_{TGS-V} : TGS와 Sever가 사전에 공유하고 있는 비밀 키, $Lifetime_1$: TGT의 유효기간, $Lifetime_2$: Ticket의 유효기간

KERBEROS

- **TGS를 사용한 인증 프로토콜 : 부가 설명**

· 부가 설명

- ③ E[Kc(TGT)] 전송과정에서 패스워드를 입력받아, 패스워드로부터 키 Kc를 만들어 복호화; 적법한 패스워드라면 올바른 Kc를 만들어 복호화가 가능하고 TGT를 획득할 수 있음
- ⑤ 서비스 승인 티켓 발행과정에서 TGS는 TGT를 복호화하여 ID_{tgs}의 존재여부, 유효기간 확인, 클라이언트의 ID와 IP주소 유효성 판단 후 티켓을 발행

KERBEROS

- **TGS를 사용한 인증 프로토콜 : 문제점**

· 문제점

- TGT의 유효기간 문제

① 유효기간이 너무 짧을 때 : 세션이 진행 중에도 패스워드를 입력해야 하는 경우가 생길 수 있음

② 유효기간이 너무 길 때 : 클라이언트가 서비스를 정상적으로 사용하고 로그아웃한 경우에도 TGT는 사용이 가능하기 때문에, 침입자가 유효기간이 남은 TGT를 이용하여 서버에 접속가능

! 티켓을 사용하고 있는 사람과 티켓을 발행받은 사람이 같음을 증명

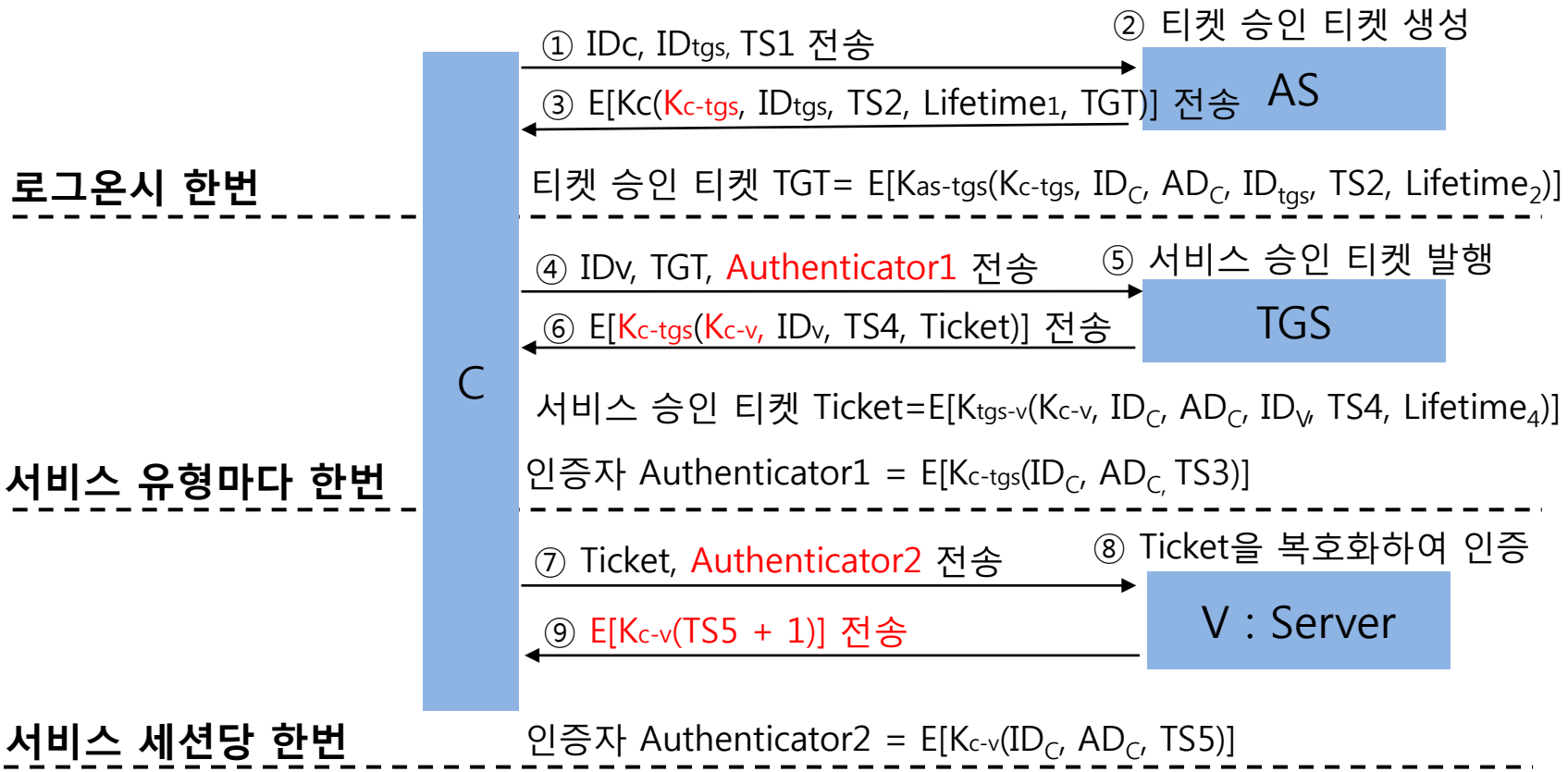
- 서버의 인증 : 가짜 서버가 사용자가 입력한 정보를 가로채서 이용할 수 있음(Spoofing)

서버의 식별자(IDv)를 위조할 수 있다면 누구든 서버 흉내를 내서 클라이언트가 보내는 메시지를 받아볼 수 있는 문제가 생김

! 서버가 서버 자신을 사용자에게 인증(상호 인증이 필요)

KERBEROS

- **Kerberos ver.4의 인증 프로토콜**



KERBEROS

- **Kerberos ver.4의 인증 프로토콜 설명**

① 로그온을 하고 TGS에 대한 접근 요청 : 클라이언트의 ID, TGS의 ID, TS1(요청 메시지를 발송한 시각)

② AS는 TGT를 생성 세션 키?? AS가 클라이언트와 TGS간의 인증에서만 쓰이는 키를 만들어서 인증에만 사용

③ AS의 응답 메시지 : 클라이언트와 TGS 사이의 세션 키와 TS2(TGT를 발송한 시각), 응답 메시지의 유효기간, TGT를 Kc로 암호화하여 전송 → 클라이언트는 Kc로 복호화하여 **TGT**를 획득

④ 서버에 대한 접근 요청 : 서버의 ID와 TGT, 인증자 전송; TGT와 인증자를 복호화하여 나온 클라이언트의 ID와 IP주소의 일치 여부를 확인하여 일치하면 Ticket 발행

⑤ 클라이언트와 서버 사이의 세션 키와 서버의 ID, TS4(Ticket을 발송한 시각), Ticket을 클라이언트와 TGS 사이의 세션 키로 암호화하여 전송 → 클라이언트는 클라이언트와 TGS 사이의 세션 키로 복호화하여 **Ticket과 클라이언트와 서버 사이의 세션 키**를 획득

⑥ 서버 접속 : 클라이언트는 서버에게 Ticket과 인증자를 전송

⑦ 응답 : 서버는 인증자에서 얻은 타임스탬프 값에 1을 더한 다음 세션 키로 암호화하여 전송

→ 서버의 상호 인증 구현

KERBEROS

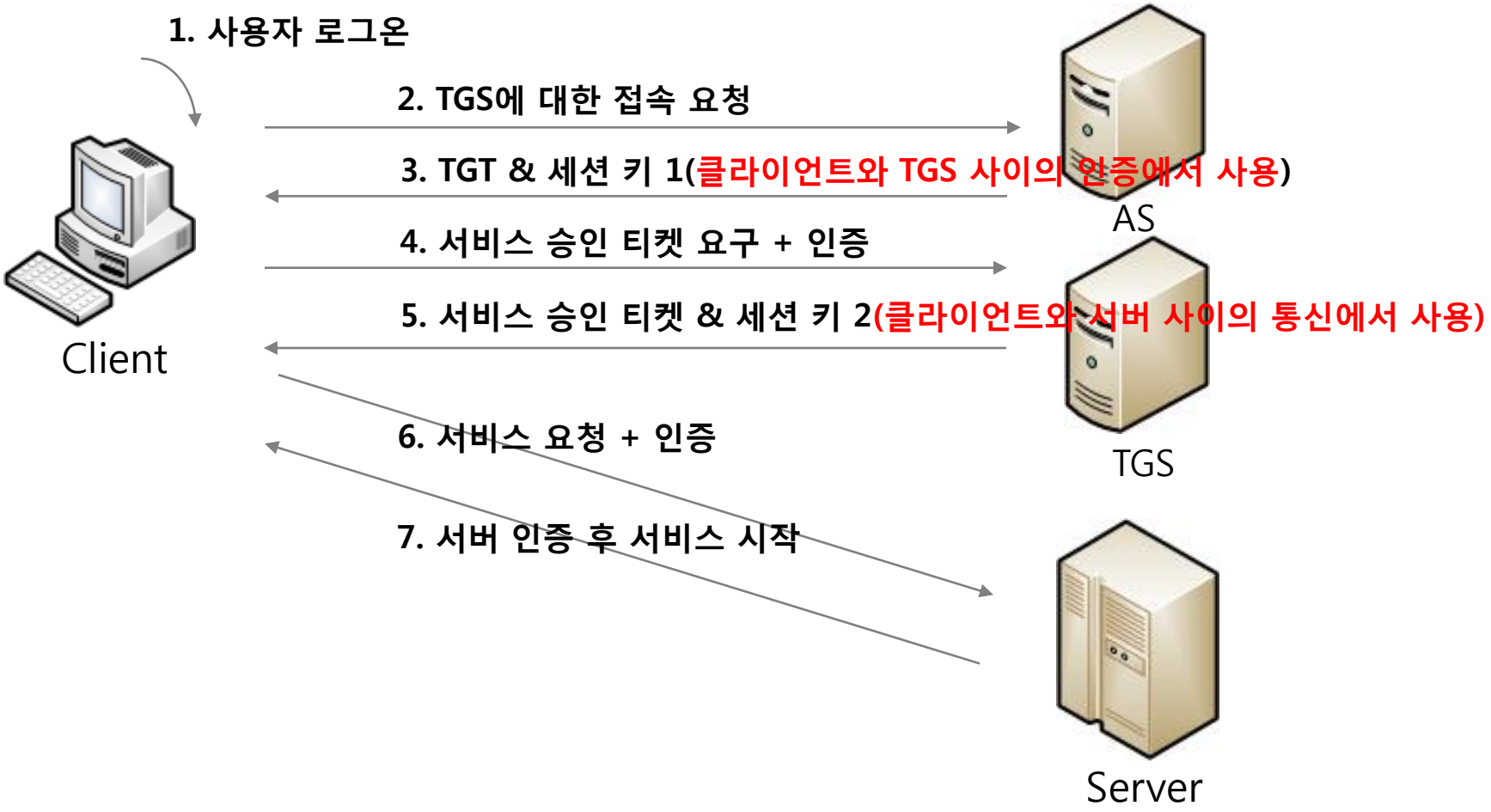
- **Kerberos ver.4의 인증 프로토콜 설명 : 사용하는 비밀 키의 분류**

  - 공유 키 : 통신 전에 사전에 미리가지고 있는 비밀 키
  - 세션 키 : 인증 또는 세션을 위해서 만들어진 비밀 키; 일회용
  - 패스워드를 통해 얻어진 키

키 분류	설명
공유 키	Kas-tgs : AS와 TGS 사이(일명 KDC 마스터 키); TGT 암호화에 사용하여 클라이언트가 임의로 변경할 수 없도록 함 Ktgs-v : TGS와 서버 사이; Ticket 암호화에 사용
세션 키	Kc-tgs : 클라이언트와 TGS 사이; TGS가 클라이언트에게 Ticket을 전해줄 때 암호화에 사용 Kc-v : 클라이언트와 서버 사이의 통신에서 사용
패스워드를 통해 얻어진 키	Kc : 클라이언트와 AS가 패스워드를 사용해서 만듬; 패스워드가 네트워크 상에 평문으로 노출되는 것을 막는 기능

KERBEROS

- **Kerberos ver.4의 인증 프로토콜(종합)**



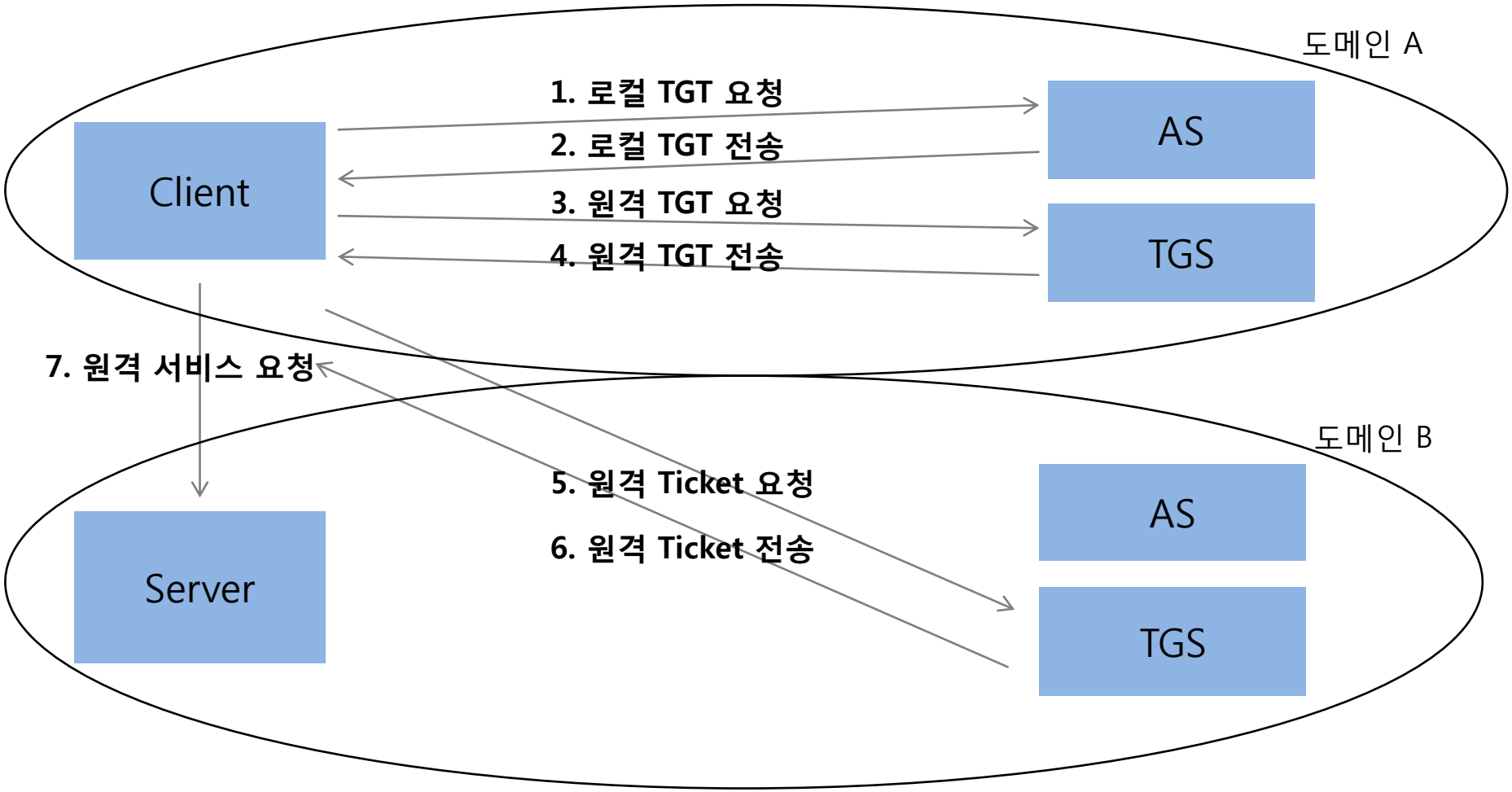
KERBEROS

- **Kerberos간의 인증**

  - **Kerberos 서버(AS+TGS), 다수의 클라이언트, 다수의 응용 서버로 구성된 환경**을 위해서는
    1. Kerberos 서버는 모든 사용자의 ID와 해시된 패스워드를 데이터베이스에 저장하고 있어야 함
    2. Kerberos 서버는 필히 각 서버와 비밀키를 공유해야 함
    3. 서로 다른 공동체 간의 Kerberos 서버는 상호간에 등록을 해야 함

KERBEROS

- **Kerberos간의 인증 : 절차**



KERBEROS

- **Kerberos ver.5 : ver.4의 결점 보완**

- 암호화 알고리즘 : DES 사용

→ Version 5는 다른 암호 알고리즘도 사용 가능

- 네트워크 주소 : IP 주소 사용

→Version 5는 다른 형식의 네트워크 주소 사용 가능

- 티켓 유효시간 : 5분 단위로 8비트 사용 가능; 최대 2^8*5 = 1280분 동안 사용 가능

→ Version 5 : 시작 시간과 끝 시간을 따로 표시하여 유효기간이 설정이 자유로움

- 클라이언트가 접근에 사용했던 인증서를 한 가지 서버에만 이용가능

→ Version 5 : 인증서를 서로 다른 서버에서도 이용 가능

KERBEROS

- **장점**

  - 인증 절차와 사용자와 서비스 간의 통신 내용을 암호화 키 및 암호 프로세스를 이용하여 보호하기 때문에 데이터의 **기밀성과 무결성**을 보장할 수 있음

- **단점**

  - 키 분배 센터에서 각 구성요소의 암호화 키를 가지고 있기 때문에, 키 분배 센터에 오류가 발생하면 전체 서비스 사용 불가능
  - 패스워드 공격에 취약 : 클라이언트를 공격 당해 패스워드가 유출되어도 KDC서버는 인지 못함