Chap 23.
Internet Authentication Applications

연세대학교

Kerberos Overview

- MIT에서 개발됨
- public domain 및 상업적 지원 버전 모두에서 사용할 수 있는 소프트웨어 유틸리티
- 인터넷 표준으로 공표되었으며 원격 인증(remote authentication)을 위한 사실상(defacto) 표준

- 신뢰할 수 있는 제3자(trusted third-party) 인증 서비스 방법을 사용
  - Kerberos가 client와 server간의 상호 인증을 중재
  - 사용자가 호출한 각 서비스(서버)에 대해 자신의 신원을 입증할 것을 요구
  - (선택사항) 클라이언트에게 서버가 자신의 신원을 입증할 것을 요구

연세대학교

Background of Kerberos Protocol

- 보호되지 않은 네트워크에서의 명백한 보안 위험 요소
  – **위장**(impersonation)
  - 이 위협에 대처하기 위해서 서버들은 서비스를 요청한 client의 신원을 확인할 수 있어야 함
- client 신원 확인 방법
  - 서버/클라이언트 간의 대화로 신원 확인
    - 서버에 부담
  - 집중화된 **인증서버**(Authentication Server: AS) 사용
    - AS는 모든 client의 패스워드 정보 보관
    - 사용자는 처음에 신원 확인을 위해서 AS에 로그인하고, 특정 서버 접근을 요청함
    - AS는 신원 확인 후에 application server로 정보를 전달하고, application server는 client로부터의 서비스 요청을 받아들임

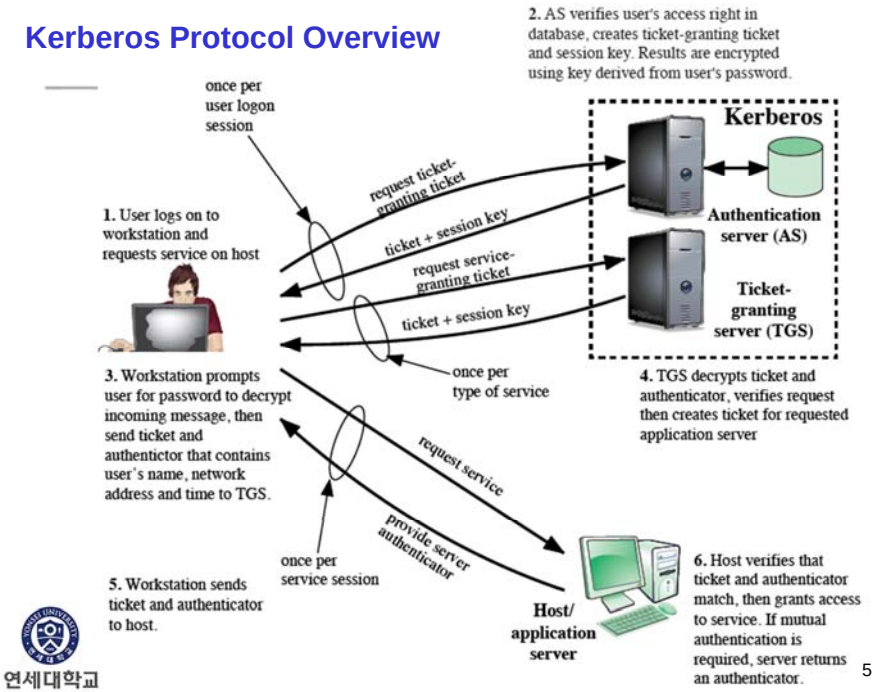
연세대학교

- 다음 위협 요소에 대처하여 신원확인 작업을 안전하게 수행하는 방법을 찾을 필요가 있음
  - client가 password를 네트워크를 통해 AS로 보낼 때에 공격자(opponent)가 그 password를 볼 수 있음.
  - 공격자가 AS를 위장하여 거짓 검증결과를 보낼 수 있음

- Kerberos 프로토콜은
  - clients, application servers, a **Kerberos** server가 참여함
    - Kerberos서버는 **인증서버**(AS)와 **티켓허가서버**(TGS)로 구성
  - 클라이언트/서버 대화의 보안에 대한 다양한 위협에 대응할 수 있도록 설계됨
  - 암호화(DES 알고리즘)와 메시지 집합을 사용함

연세대학교

Kerberos Protocol Overview

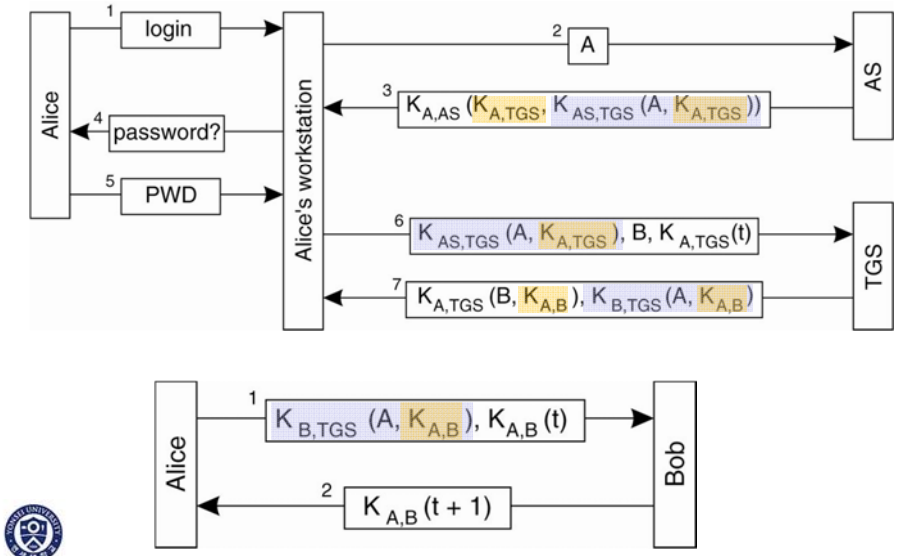


Kerberos Protocol

- User client-based login
  - 사용자가 workstation에 로그인하고, 특정 서비스 접근 요청
  - 사용자ID와 ticket-granting ticket(TGT)요청 메시지를 AS로 전송
- Client authentication
  - AS는 자신의 database를 조회하여 신원을 확인하고 password를 찾음 → password는 비밀키 생성에 사용
  - ticket(TGT)과 session key를 client에게 **암호화** 전송
    - TGT는 사용자ID, 네트워크주소, 유효기간 등을 포함하며, AS와 TGS의 공유 비밀키로 암호화되어 client가 변경할 수 없음
  - client는 패스워드를 사용하여 수신 메시지를 복호화하여 TGT와 session key를 복구함
  - 네트워크를 통해 password 전송을 하지 않는 방식을 사용

- Client service authorization
  - client는 service-granting ticket 요청 메시지를 TGS로 전송
    - TGT와 서비스ID, 암호화된 authenticator(사용자ID, 네트워크주소, 유효기간)
  - TGS는 TGT와 인증자(authenticator)를 복호화하여 연관된 수명과 client에 대한 티켓허가를 확인함
  - TGS는 service-granting ticket과 session key를 client에게 전송
  - TGT는 새로운 서비스 요청에 재사용 가능, 인증자는 새로 생성
- Client service request
  - client는 service-granting ticket과 암호화된 인증자를 application server로 전송
  - 상호인증이 필요하면 server는 timestamp를 1 증가시켜 암호화하여 응답

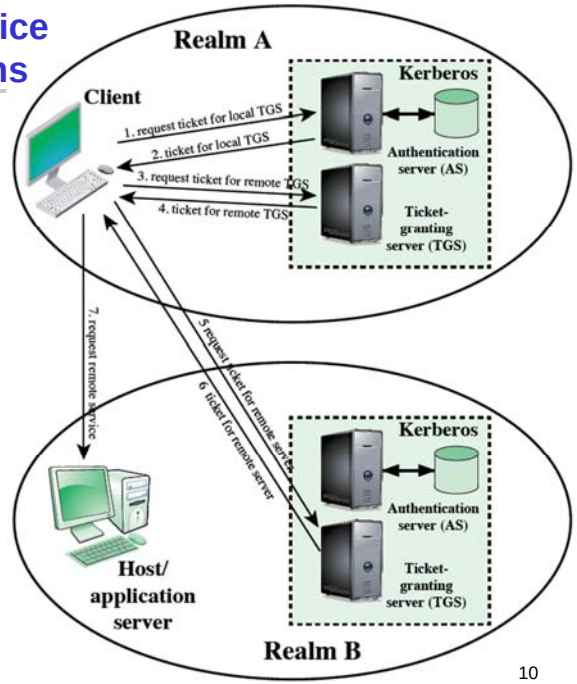
Authentication and Setting up Secure Channel



Kerberos Realms

- Kerberos 환경은 다음으로 구성됨
  - a Kerberos server
  - 다수의 client - kerberos server에 등록(패스워드)
  - 다수의 application server – kerberos server와 비밀키를 공유
- 이러한 환경을 Kerberos 영역(realms)이라고 함
  - 서로 다른 관리 조직에 속한 클라이언트와 서버의 네트워크는 일반적으로 서로 다른 영역을 구성함.
- 복수의 Kerberos 영역이 있는 경우:
  - Kerberos 서버들은 비밀키를 공유해야 하고, 사용자를 인증하기 위해 다른 영역의 Kerberos 서버를 신뢰해야 함.
  - 두 번째 영역에 참여하는 서버는 첫 번째 영역에 있는 Kerberos 서버를 신뢰해야 함.

Request for Service in Another Realms



Kerberos Versions 4 and 5

- Kerberos v4 – 가장 널리 사용되는 Kerberos 버전
- 버전 5에서 향상된 기능
  - 암호화된 메시지는 암호화 알고리즘 식별자로 태그됨
    - 사용자가 DES 이외의 알고리즘을 사용하도록 Kerberos를 구성 할 수 있습니다.
  - 인증 전달(forwarding)을 지원
    - 클라이언트가 서버에 액세스하여 해당 서버가 클라이언트를 대신하여 다른 서버에 액세스하게 함.
    - 버전 4에서 보다 적은 수의 보안 키 교환을 필요로 하는 영역 간 인증을 위한 방법을 지원

Kerberos Performance Issues

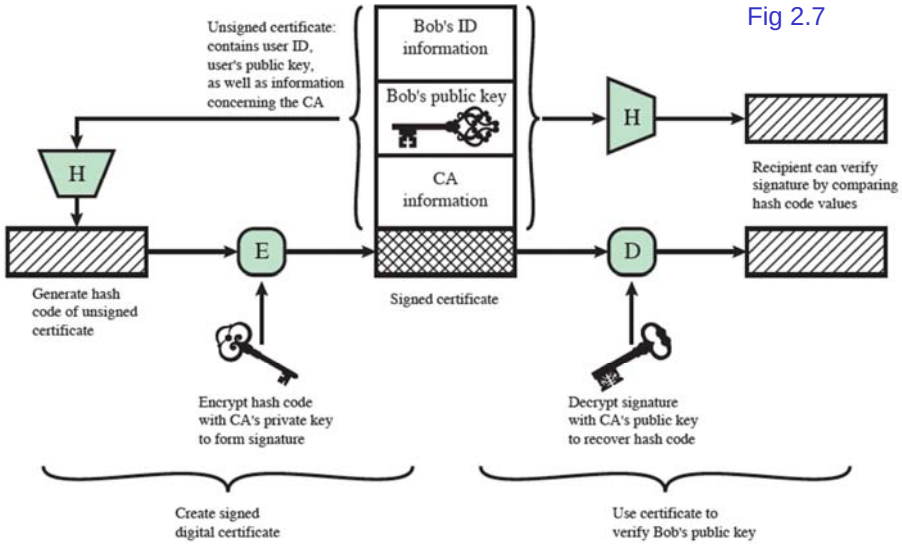
- client-server 설치가 확대되고, 네트워크 환경의 규모가 커짐
  - 로그온 인증이 더욱 중요해짐
- 대규모 환경에서의 Kerberos 성능 영향
  - 시스템이 적절히 구성되어 있다면 영향은 거의 없음
    - ticket을 재사용하여 트래픽을 줄임
- Kerberos 보안
  - Kerberos 서버를 분리된 독립적인 컴퓨터에 배치할 때가 최선임
- 복수의 Kerberos 영역에 대한 동기
  - 관리적인 측면에 있으며, 성능과 관련 없음

Certificate Authority (CA)

- 인증서(Certificate)의 구성
  - 공개키 + 사용자ID
  - 신뢰할 수 있는 제3자의 서명
- 인증기관(Certificate Authority: CA)
  - 신뢰할 수 있는 제3자는 일반적으로 사용자 커뮤니티가 신뢰하는 인증기관(CA)임 (예: 정부기관, 금융기관 등)
- 사용자는 자신의 공개키를 안전한 방식으로 인증기관에 제시하고 인증서를 얻을 수 있음
  - 사용자는 인증서를 게시(publish)할 수 있음.
  - 이 사용자의 공개키가 필요한 사용자는 인증서를 가져와서 첨부된 **서명**을 통해 인증서가 유효한지 확인할 수 있음.

Public Key Certificate

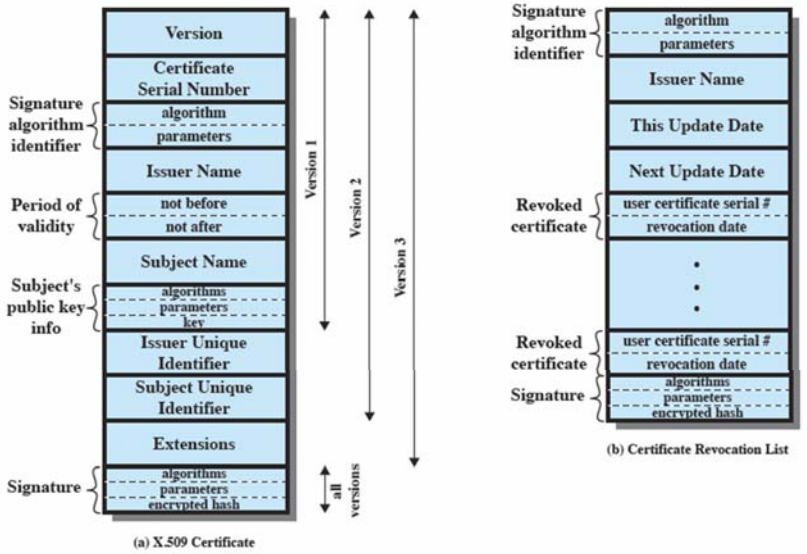
Fig 2.7



X.509

- X.509
  - 공개키 인증서 형식을 정의하는 ITU-T 표준
  - RFC 5280으로도 지정됨
  - 가장 널리 사용되는 공개키 인증서 형식 (현재 v3 사용)
  - CCITT(ITU-T) X.500 directory service standards의 부분
- 인증서는 대부분의 네트워크 보안 응용 프로그램에서 사용
  - IP security (IPsec)
  - SSL (Secure Sockets Layer)
  - S/MIME 등
  - Secure electronics transaction (SET)
  - eBusiness applications
- 공개키 암호화와 디지털 서명을 사용
  - 알고리즘이 표준으로 지정되지 않았지만 RSA를 추천

X.509 Certificates



- X.509 공개키 인증서
  - subject(키소유자) name, 발행자(CA) name, 공개키정보, 유효기간 등을 포함하고 발행자가 서명
- 인증서 취소 리스트 (Certicate revocation list: CRL)
  - 발행자 이름과 취소된 인증서들의 serial number와 취소날짜를 포함하고 발행자가 서명
  - application은 인증서를 받을 때에 CRL을 검사하여 인증서 취소 여부를 결정해야 함 – 많은 application에서 오버헤드 때문에 이 과정을 실행하지 않음
  - 대체 방안 - Online Certificate Status Protocol (OCSP) 사용
    - 특정 인증서가 유효한 지 CA에게 질의

Specialized variants of Certificates

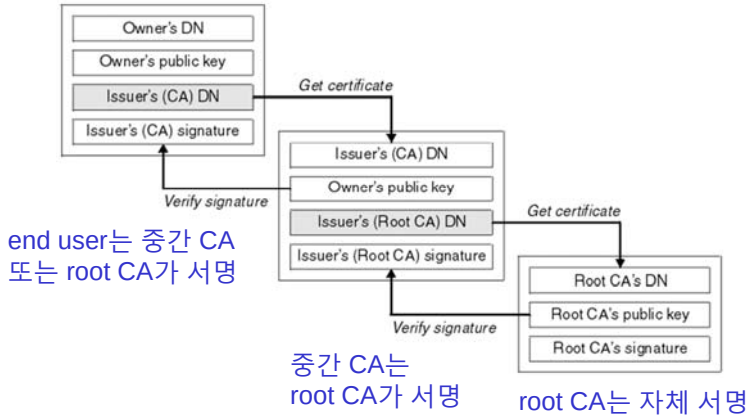
- Conventional (long-lived) certificates
  - CA 인증서와 “end user” 인증서
    - CA 인증서는 다른 인증서 서명용으로만 사용
    - end user 인증서는 신원확인, 여러 응용에서의 서명용으로 사용
  - 유효기간 : 일반적으로 수개월에서 수년간
- Short-lived certificates
  - 기존 인증서의 오버 헤드 및 제한을 피하면서 그리드 컴퓨팅과 같은 응용 프로그램에 인증을 제공하는 데 사용
  - 유효기간 : 수 시간에서 수 일간
  - 일반적으로 공인 CA가 발급하지 않기 때문에 발급 기관 외부에서 검증하는 데 문제가 있음

- Proxy certificates (대행 인증서)
  - 단기 인증서의 일부 제한을 해결하면서 그리드 컴퓨팅과 같은 응용 프로그램에 인증을 제공하는 데 널리 사용 (RFC 3820)
  - "proxy certificate" 확장 기능의 존재 여부로 식별되며 end user 인증서가 다른 인증서를 서명 할 수 있게 함.
  - 사용자가 전체 인증서를 제공하지 않고도 일부 자원 접근에 대한 자격 증명(credential)을 쉽게 만들 수 있도록 함.
- Attribute certificates
  - 사용자 ID를 권한 부여 및 접근 제어에 사용되는 속성 집합에 연결하기 위해 다른 인증서 형식을 사용함 (RFC 5755)
  - 사용자는 다른 목적을 위해 서로 다른 속성 집합을 가진 여러 가지 속성 인증서를 가질 수 있음
  - "Attributes"확장에 정의됨

Public-Key Infrastructure (PKI)

- 공개키 기반구조(PKI) – RFC 4949
  - 비대칭 암호화를 기반으로 디지털 인증서를 생성, 관리, 저장, 배포 및 취소하는데 필요한 하드웨어, 소프트웨어, 사람, 정책 및 절차 집합
  - **목적**: 공개키의 안전하고 편리하며 효율적인 획득을 가능하게 하기 위해 개발됨
- trust store (신뢰저장소)로 구현됨
  - CA와 그들의 공개키들의 커다란 리스트를 저장
  - 문제
    - 인증서 입증에 문제가 있을 때에 사용자에게 결정을 맡김
    - 신뢰저장소의 모든 CA를 동등하게 신뢰함
      - 실제로는 신뢰할 수 없는 것이 포함될 수 있음 아닐 수 있음
    - 웹 브라우저, 운영체제에서의 다른 구현이 다른 신뢰저장소를 사용하며 사용자들에게 다른 보안관점을 제시

Certificate Chain

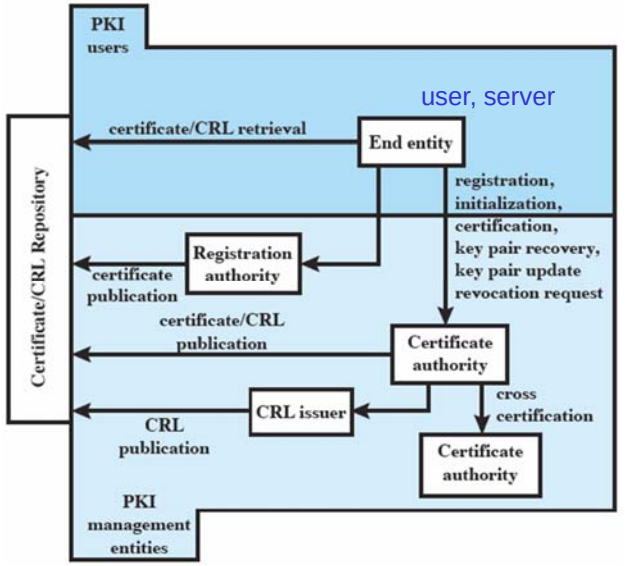


end user는 중간 CA
또는 root CA가 서명

중간 CA는
root CA가 서명

root CA는 자체 서명

Public Key Infrastructure (PKIX)



PKIX Management Functions

- 등록(registration)
- 초기화(initialization) – 키 요소의 초기화
- 인증(certification) – CA가 인증서 발행
- 키 쌍 복구 및 갱신
- 인증서 취소 요청
- CA들 간의 상호 인증