정보학연구 제6권 제1호 2003년 3월(pp.107-115)
The Journal of Information Technology, Vol.6, No.1

분산시스템에서의 속성인증서를 이용한 접근통제방안

김지홍*, 박종화**, 황태현***

요 약

정보통신기술의 발달로 인터넷상의 공개키 인증서를 사용한 전자거래가 활성화되고 있다. 그러나 실제적으로 웹 서버, 데이터베이서 서버에 접속하기 위한 접근통제를 위한 방안으로 속성인증서에 대한 연구도 활발히 진행되고 있다. 본 논문에서는 기존의 속성인증서에 대한 적용방식을 분석하고, 이를 이용하여 분산시스템에서 속성인증서를 이용한 다중 응용서버에 대한 접근통제방안을 제시한다.

Ⅰ. 서론

정보통신 기술의 발달로 사회의 모든 분야에서 인터넷의 활용이 급속히 확산되어 전자결제, 전자상거래, 인터넷 뱅킹 등의 편리한 서비스가 제공되고 있다. 그러나 인터넷을 이용한 모든 거래는 거래 당사자간 비접촉, 비대면을 특징으로 하기 때문에, 온라인상의 편리함을 추구할 수 있는 반면에 거래 당사자간의 상호신뢰에 있어서 취약성을 가진다. 이러한 단점을 해결하기 위하여 전세계적으로 공개키기반구조(PKI: Public Key Infrastructure)라는 인증기반 구조[7]를 도입함으로서 거래당사자들 간의 신뢰성과 안전성을 추구하고 있다. 공개키기반구조는 계층구조 형식의 인증기반구조를 채택함으로서, 하위 계층의 인증기관 혹은 사용자에게 공개키 인증서를 발급하고, 이를 이용하여 온라인상의 안전한 전자거래를 할 수 있도록 하는 방식이다. 따라서 공개키기반구조 상에서의 모든 사용자는 공인인증기관으로부터 사용용도에 부합되는 공개키 인증서를 발급받고, 이를 이용하여 자신이 정당한 사용자임을 입증할 수 있다.

그러나 이러한 공개키 인증서를 이용한 기술은 공개키 정보를 이용하여 사용자 인증정보를 제공하므로 비대면 인터넷 통신에서의 사용자 신원을 입증하기 위해서 유용하게 사용될 수 있지만, 실제 시스템에서의 접근통제를 위한 정보는 포함하고 있지 않으므로, 접근통제를 필요로 하는 분야에서는 속성인증서와 같은 별도의 형태의 인증서를 이용한 구조가 제안되고 있다.

속성인증서는 사용자의 속성정보를 저장하는 인증서로서, 사용자의 지위, 권한, 임무 등과 같은 다양한 권한정보를 제공한다. 속성인증서에 대한 연구는 ITU-T, IETF 등에서 진행되고 있으며, IETF에서는 Internet Draft 문서[4]와 RFC 3281[6]를 통하여 표준화가 진행되고 있다. 이러한 속성인증서는 사용자의 속성정보와 같은 유용한 정보를 저장하고 있지만, 사용자에 대한 공개키 정보를 가지고 있지 않다. 따라서 속성

* 세명대학교 정보보호학과
** 세명대학교 소프트웨어학과,
*** 한국전자통신연구원 컴퓨터소프트웨어연구소

인증서를 접근통제 분야에 적용하기 위해서는 공개키 기반구조상의 PKI 인증서를 첨부하여 접근하거나, 혹은 PKI 인증서와 속성인증서를 결합한 형태에 관한 많은 연구가 진행되고 있다 [2,3].

본 논문은 이와같은 속성인증서에 대한 기존 연구에 대한 설명과 함께, 이들의 장단점을 분석하고, 실제 응용분야에 적용할 수 있는 속성인증서를 이용하여 웹서버를 통한 다양한 응용서버에 대한 분산시스템에서의 접근통제방안을 제시한다.

II. 기초이론

2.1. 속성인증서

인증서(certificate)란 여권과 같이 자기 자신의 신분을 증명하기 위해 사용되는 증서로서, 인터넷상에서 신뢰성있는 통신을 위하여 개개인을 입증하기 위해 사용된다. 이와같이 인터넷상에서 사용되는 인증서는 크게 PKI 인증서, 속성인증서, SPKI 인증서로 분류할 수 있다. PKI 인증서란 현재 범용적으로 사용되고 있는 공개키 기반구조에서 사용되고 있는 공개키 인증서를 말한다. 본 논문에서는 PKI 인증서[7]와 SPKI 인증서에 관한 설명은 생략한다.

속성인증서(Attribute Certificate)는 사용자의 속성정보를 저장하는 인증서로서, 사용자의 지위, 권한, 임무 등과 같은 다양한 권한정보를 제공하며, 속성인증서에 대한 표준에는 IETF (Internet Engineering Task Force)와 ITU-T (Intenational Telecommunication Union)가 있다.

IETF에서 제안된 속성인증서 표준은 RFC 3281[6]이며, ITU-T와 ISO/IEC가 함께 제안한 X.509 2000년 버전으로, ISO/IEC 9594-8에 명시된 속성인증서 프로화일이 있다. 두 개의 표준은 상당부분 유사한 구조를 가지고 있으며, IETF에서 제안된 속성인증서에 대한 형식[6,7]은 표 2-1과 같다.

〈표 2-1〉 속성 인증서 형식

기본영역	사용용도
버전	X.509 V2.0인 경우 "2"
사용자(holder) 이름	속성인증서 사용자이름(X.500 이름)
발급자(issuer) 이름	속성인증서 발급자이름(X.500 이름)
서명알고리즘	서명알고리즘ID 및 관련파라미터
일련번호	속성인증서 일련번호
유효기간	시작일자와 만료일자
속성정보	사용자의 속성정보
발급자 고유ID	발급자에 대한 부가정보
확장자	속성인증서에 대한 부가정보
서명문	인증서발급자의 서명문

속성인증서의 구성은 기본적으로 PKI 인증서 형식과 유사하다. 그러나 사용자의 공개키를 포함하고 있지 않으며, 사용자의 속성정보는 다음과 같다.

- Service Authentication Information : 사용자 ID 및 비밀번호
- Access Identity : 속성인증서 holder에 대한 정보
- Charging Identity : 과금을 위한 정보
- Group : 사용자가 속한 그룹
- Role : roleAuthority, 사용자의 역할
- Clearance : 보안인가등급

속성인증서는 서버 혹은 데이터베이스 등 시스템 자원에 대한 접근통제를 목적으로 하기 때문에, 인증서 발급주기를 가급적 짧게 하고,

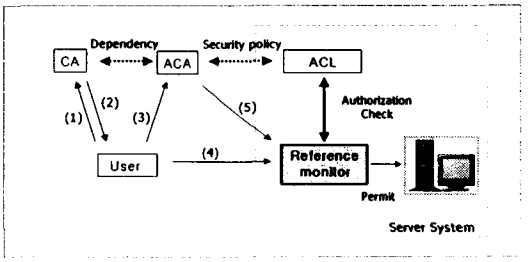
CRL(Certificate Revocation List)은 가능하면 사용하지 않는 것을 권장하고 있다.

2.2 속성인증서를 이용한 응용서버 접속방법

속성인증서 분배 방식으로 서버 Pull 방식과 사용자 Pull 방식이 있다.

1) 서버 Pull 방식을 이용한 접근제어 모델

<그림 2-1>은 서버 Pull 방식을 이용한 서버 접속방식을 도식화하고 있으며, 이와같은 절차는 다음과 같다.



〈그림 2-1〉 서버 Pull 방식을 이용한 속성인증서 획득방법

① 사용자는 인증기관(CA)으로부터 신원확인을 통해 공개키 인증서를 요청한다.

② 인증기관에서는 사용자의 신원확인, 키발급절차를 거쳐, 사용자에게 공개키 인증서를 발급 받는다.

③ 사용자는 속성인증기관(ACA)으로부터 속성인증서를 요청한다. 이때 사용자는 자신의 공개키인증서를 첨부하여 신원확인을 한다. 사용자의 접근권한 확인과정을 거친 후, 발급된 속성인증서는 별도의 속성인증서 저장소에 보관된다.

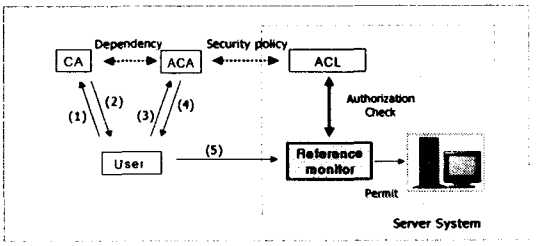
④ 사용자가 서버에 접속하는 경우, 자신의 공개키인증서를 제출한다.

⑤ 서버에서는 사용자의 공개키인증서를 받고, 속성인증서버에 접속하여 사용자에 대한 속성인증서를 획득한다.

이와 같은 절차를 통하여 속성인증서에 명시된 사용자의 접근권한을 ACL을 통하여 확인하고, 서버접근을 허용하거나 접속을 거부한다.

2) 사용자 Pull 방식을 이용한 접근제어 모델

<그림 2-2>는 사용자 Pull 방식을 이용한 접근제어 모델을 도식화하고 있으며, 이와같은 절차는 다음과 같다.



〈그림 2-2〉 사용자 Pull 방식을 이용한 속성인증서 획득방법

① 사용자는 인증기관(CA)으로부터 신원확인을 통해 공개키 인증서를 요청한다.

② 인증기관에서는 사용자의 신원확인, 키발급절차를 거쳐, 사용자에게 공개키 인증서를 발급 받는다.

③ 사용자는 속성인증기관(ACA)으로부터 속성인증서를 요청한다. 이때 사용자는 자신의 공개키인증서를 첨부하여 신원확인을 한다.

④ ACA에서는 사용자의 접근권한 확인과정을 거친 후, 발급된 속성인증서는 사용자

에게 전달된다.

⑤ 사용자가 서버에 접속하는 경우에는 자신의 공개키인증서와 속성인증서를 제출한다.

이와같은 절차를 통하여 서버에서는 속성인증서에 명시된 사용자의 접근권한을 ACL을 통하여 확인하고, 서버접근을 허용하거나 접속을 거부한다.

3.3. 속성인증서 분배 방식에 대한 비교 설명

사용자 Pull 모델은 사용자가 서버시스템에 접근하기 위하여, 자신의 속성인증서를 직접 전달하는 방식을 말한다. 서버 Pull 모델에서는 속성인증서 보관을 위해 별도의 디렉토리를 사용하는 방식이다.

일반적으로 사용자 Pull 모델은 속성인증서 검색을 위한 추가적인 통신·설치비용이 필요하지 않기 때문에 속도가 향상된다는 장점을 가지고 있으나, 속성인증서의 관리상 분실과 훼손 혹은 전달과정에서 해커나 크래커에 의해 도청될 수 있는 단점이 있다. 또한 속성인증서의 짧은 생명주기 측면을 고려할 때, 사용자가 보관하는 것보다, 신속한 갱신과정을 수행할 수 있도록 별도의 저장소를 사용하는 방식이 선호된다.

2.4. Kerberos 프로토콜을 이용한 인증방법

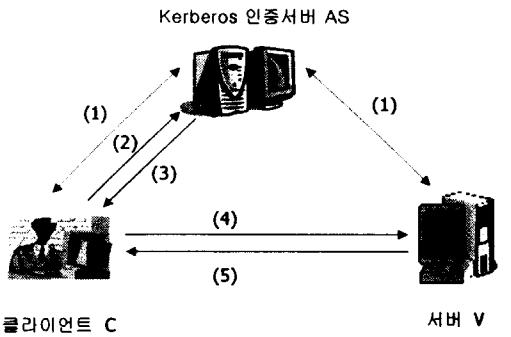
Kerberos 프로토콜[1]는 MIT Arena 프로젝트를 통해 분산시스템에서의 인증서비스를 제공한다. Kerberos 프로토콜은 사용자인 클라이언트 C(Client)와 검증자인 서버 V(Server), 그리고 신뢰할 수 있는 제 3자인 Kerberos 인증서버 AS(Authentication Server)로 구성된다. 초기 설정시에는 C와 AS, V와 AS간의 비밀키를 공유한다. Kerberos 프로토콜에서 사용되는 시스템 계수는 다음과 같다.

C : 클라이언트
AS : 인증서버
V : 서버
IDc : 사용자 식별자
Pc : 사용자 패스워드
IDv : 서버 V의 식별자
E : 대칭키 암호알고리즘
ADc : 사용자 네트워크 주소

이와 같은 시스템계수를 이용하여 간단한 Kerberos 프로토콜은 <그림 2-3>의 과정을 통하여 인증기능을 수행한다.

〈표 2-1〉 속성인증서 분배 방식 비교

	사용자 Pull 방식	서버 Pull 방식
장점	- 서버 부하가 줄어듬 - 속성인증서 관리를 위한 저장소 설치비용 및 검색을 위한 통신비용이 필요하지 않다.	- 속성정보 변경에 유연성을 가진다. - 속성인증서 전달, 보관 과정에서 발생하는 문제점을 줄일 수 있다.
단점	- 속성인증서 전달, 관리측면에서 분실 및 훼손의 여지가 있다.	- 속성인증서 관리를 위해 추가적인 비용이 필요하다. - 서비스 사용자의 수에 비례하여 통신 부하가 발생한다.



〈그림 2-3〉 Kerberos 인증기법

(초기설정단계)

① C와 AS는 비밀키 $K_{C,AS}$를 공유하며, S와 V는 $K_{S,v}$를 공유한다.
TicketV = $E_{KS,V}(ID_C \| AD_C \| ID_V)$로 정의한다.

(인증단계)

② C->AS : $ID_C \| P_C \| ID_V$
사용자 C는 AS에게 사용자의 ID와 패스워드, 그리고 서버V의 ID를 AS에게 전송한다.

③ AS->C : TicketV
인증서버 AS는 사용자 C에게 서버 V에 접속할 수 있는 티켓 TicketV를 발행한다.

④ C->V : IDc || TicketV
사용자 C는 AS로부터 수신된 티켓 TicketV를 가지고 서버 V에 접근한다.

⑤ V : 검증작업
검증자 서버 V는 TicketS를 복호하여, 적법한 요구이면 접근을 허용한다.

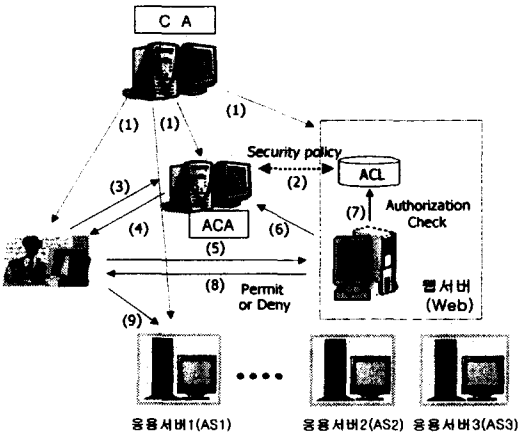
일반적으로 Kerberos 프로토콜은 안전성(secure), 신뢰성(reliable), 투명성(transparent), 규모적응성(scalable) 이 공개적으로 입증된 프로토콜로서, 분산시스템에서의 응용서버 보안 대책으로 널리 사용되고 있다.

Kerberos 버전 4에서는 Kerberos 서버에 인증서버(AS : Authentication Server)와 티켓발행서버(TGS : Ticket Generating Server)를 두고, AS에서 TGS서버에 접속하기 위한 $Ticket_{TGS}$을 발행받고, 다시 TGS 서버에 $Ticket_{TGS}$ 을 이용하여 접속하여 인증후, 서버에 접속하기 위한 $Ticket_v$ 을 발행받고, 다시 $Ticket_v$를 이용하여 인증후, 서버에 접속하는 과정으로 수행된다.

Ⅲ. 제안 방법

Kerberos 프로토콜이 성립되기 위한 사전단계로서 비밀키 공유가 필요하다. 그러나 본 논문에서는 공개키기반구조를 이용함으로서, 기본적으로 상대방의 공개키를 이용하여 암호화통신을 사용할 수 있도록 하였다.

전 절에서 제시된 속성인증서 전달방법에 대한 내용을 분산시스템에 적용하기 위하여 커버로스 인증방법을 적용하였을 경우에 제안한 방법은 <그림 3-1>과 같다.



〈그림 3-1〉 분산시스템에서의 웹서버를 통한 접근제어

(사전단계)

① 공개키 인증서 발급단계 :
사용자, 속성인증서 발급기관(ACA), 웹서버, 응용서버들은 공개키기반구조상의 인증기관으로부터 인증서를 요청하고, 신원확인과정을 거친 후에 공개키인증서를 발급받는다.

② ACA 보안정책 설정 :
관리자는 웹서버를 통하여 접근할 수 있는 응용서버들에 대한 접근제어 정책을 설정한다. 즉, 각 응용서버들에 대한 접근통제 리스트인 ACL(Access Control List)을 작성하고, 이에 대한 정보를 속성인증서 발급기관에서 보관한다.

(속성인증서 발급단계)

③ 속성인증서 요청단계 :
사용자는 접근하고자 하는 응용 서버에 대한 속성인증서를 발급하는 속성인증기관(ACA)에 접속하여 자신의 공개키인증서(Cert_U)와 속성인증서 요청패킷(Req_AC)을, ACA의 공개키를 사용하여 암호화하여 속성인증서를 요청한다.

U-> ACA : Cert_U || E_{Kp_ACA}[Req_AC]

④ 속성인증서 발급단계 :
ACA는 Cert_U를 이용하여 사용자의 신원을 확인하고, 사용자로부터 수신된 E_{Kp_ACA}[Req_AC]를 자신의 개인키를 이용하여 사용자의 속성인증서 요구패킷을 추출한다. 사용자의 요구패킷에는 사용자가 접속하고자하는 서버와 응용서비스에 대한 요구내용이 포함된다.
ACA 에서는 시스템관리자가 설정한 보안정책을 검토하여, 사용자의 요구가 적합한지를 확인하고 적법한 경우에는 속성인증서를 발급하여 이를 사용자의 공개키(E_{Kp_U})로 암호화하고, 자신의 공개키인증서(Cert_ACA)와 함께 사용자에게 전달한다.

ACA -> U : Cert_ACA ||
E_{Kp_U}[ACert_U]

(웹서버 접근단계)

⑤ 사용자는 자신의 공개키인증서(Cert_U)와 응용 서버1에 접속하여 사용하고자 하는 서비스 요구패킷[ACert_U || Req(AS1_SVC)]을 웹서버의 공개키로 암호화하여 웹서버에 보낸다. 여기서 ACert_U는 ACA의 공개키이며, Req(AS1_SVC)은 AS1에 대한 서비스 요구 패킷이다.

U-> Web : Cert_U || E_{Kp_wsv}[ACert_U ||
Req(AS1_SVC)]

⑥ 웹서버는 사용자의 공개키(Cert_U)를 확인하고, 사용자의 서비스요구를 확인하기 위하여 ACA에 접속하여 사용자의 속성인증

서의 유효성과 적법성을 검증한다.

⑦ 웹서버는 사용자의 속성인증서를 검증한 후, 적법한 서비스요구이면 사용자의 요구에 맞는 서비스 티켓 [Ticket_(AS1_SVC || TS1)]을 응용서버1의 공개키로 암호화 {E_{Kp_as1} [Ticket_(AS1_SVC || TS1)]} 하고 이를 사용자의 공개키로 다시 암호화하여 발부한다. TS1은 웹서버에서 발행한 Time Stamp로서 해당서비스에 대한 사용시간을 제한하기 위해 사용된다.

Web->U : Cert_WSV || E_{Kp_u} {E_{Kp_as1} [Ticket_(AS1_SVC || TS1)]}

(응용서버1 접근단계)

⑧ 사용자는 웹서버로부터 발급받은 티켓을 이용하여 응용서버1에 접속하고, 서비스를 사용한다.

U->AS1 : Cert_U || E_{Kp_AS1} [Ticket_(DB1_SVC || TS1)]

Ⅳ. 제안 방식의 특징

본 논문에서 제안한 분산시스템구조에서의 속성인증서를 이용한 접근통제방안은 기존의 제안된 방식에 비해 다음과 같은 특징을 가진다.

1) 분산시스템에서의 웹서버 역할

여러 종류의 응용서버를 포함한 분산시스템에서 웹서버는 사용자에게 발행된 속성인증서를 이용하여 해당 응용서버에 접근할 수 있는 티켓을 발부하는 형태로 구성된다.

2) 공개키인증서 방식을 Kerberos 프로토콜에 적용

Kerberos 프로토콜에서는 사용자와 Kerberos 서버, 웹서버와 Kerberos 서버, 응용서버와 Kerberos 서버간의 비밀키를 공유하여야 하며, ID와 패스워드를 사용하는 방식이지만, 본 논문에서는 공개키인증기관을 이용하여 모든 개체들이 공개키를 가지고 있으므로, 별도의 비밀키 공유과정이 필요없으며, 또한 비밀키를 관리하기 위한 관리기술이 필요없다.

3) 안전성이 입증된 Kerberos 방식 적용

일반적으로 Kerberos 프로토콜은 안전성(secure), 신뢰성(reliable), 투명성(transparent), 규모적응성(scalable) 이 공개적으로 입증된 프로토콜로서, 분산시스템에서의 응용서버 보안 대책으로 널리 사용되고 있다. 본 논문에서는 이와같은 Kerveros 프로토콜을 속성인증서를 이용한 접근통제에 적용함으로서, 공개키인증서와 속성인증서를 이용한 분산시스템에서의 적용방안을 제시하였다.

Ⅴ. 결론

최근 공개키기반구조에 대한 응용분야가 확대되면서, 공개키인증서의 사용이 보편화되고 있다. 그러나 실제로 접근통제를 필요로 하는 응용분야에서는 속성인증서에 대한 필요가 급격히 증가되고 있으며, 현재 이에 대한 연구가 활발히 진행되고 있다.

본 논문에서는 분산시스템상에서 속성인증서를 이용한 접근통제분야에 실제로 적용될 수 있는 모델을 제시하였다. 제안된 모델은 분산시스

템의 서버인증을 위하여 검증된 Kerberos 프로토콜을 도입하였고, 기존의 ID와 Password 방식에 비해 보다 효율적인 키 공유 및 관리기술인 공개키기반구조를 도입하여 제안함으로서, 속성인증서를 통한 공개키기반구조의 접근통제 한계성과 Kerberos 프로토콜에서의 비밀키공유 및 관리상의 문제점을 해결하고, 분산시스템에 적용하기 위한 방안을 제시하였다. 마지막으로 본 논문을 계기로 속성인증서를 이용한 프로토콜에 대한 연구가 더욱 더 활발히 진행될 수 있을 것으로 기대한다.

참고문헌

[1] William Stallings, "Network and Internetwork Security". Prentice Hall, 1998.

[2] Joon S. Park and Sandhu, "Smart Certificates: Extending X.509 for Secure Attribute Service on the Web", NISSC / 1999

[3] Joon S. Park and Sandhu, "Binding Identities and Attributes Using Digitally Signed Certificates", ACSAC / 2000

[4] "X.509_4th Edition Draft V8 - Draft ISO/IEC 9594-8", May 3. 2001.

[5] Internet Draft "An Internet Attribute Certificate Profile for Authorization", S.Farrel, June 2001.

[6] RFC 3281 "An Internet Attribute Certificate Profile", S. Farrell, April 2002.

[7] RFC 2459, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", IETF PKIX Working Group, January, 1999.

Access Control using Attribute Certificates in Distributed System

Ji-Hong, Kim*, Chong-Hwa, Park**, Tae-Hyun, Hwang***

Abstract

With the development of Information Communication Technique, electronic commerce is widely used in internet using public key certificates. And the study for access control in database system is also progressed actively. In this paper, we analyze access control mechanism using attribute certificates and we propose new access control mechanism in distributed system using attribute certificates.

* Department of Information Security, Semyung University
** Department of Software, Semyung University
*** Electonics and Telecommunications Reserch Insitite