通信情報保護學會誌
第5卷 第3號, 1995. 9

디렉토리 모델과 정보보호 서비스+

The Models and Security Services of Directory System

최 용 락*, 강 창 구**, 김 대 호**

요 약

X.500 디렉토리 표준은 전세계적인 규모의 정보통신망을 상호 연결하여 다목적 분산 디렉토리 서비스를 구축하기 위한 기초를 제공한다. 본래의 디렉토리 표준은 ITU에서 1988년 X.500시리즈로 발표되었고, 1992년에 확장 개선된 표준이 제정되어 계속 그 적용 분야를 넓혀가고 있다. 본 고에서는 디렉토리 모델에 대한 일반적 기능을 살펴보고 인증 골격과 액세스 제어를 중심으로 한 정보 보호 서비스에 관하여 고찰하였다.

1. 서 론

최근 정보산업의 급격한 발전에 따라 정보산업육성을 위한 대규모 전산망계획이 국가적 차원에서 진행되고 있다. 우리나라의 초고속 정보통신망 계획을 비롯하여 미국의 NII (National Information Infrastructure), 유럽의 Big Bang, 일본의 신사회 자본, 싱가폴의 TI2000등 초고속 정보통신망 계획을 통하여 국민생활 향상 및 국가경쟁력 강화에 힘쓰고 있다[1]. 그러나 초고속망을 이용한 서비스에서 데이타의 도용, 변조, 파괴 또는 주요 시스템의 불법적 사용등 안전 대책이 지원되지 않을 경우 초고속 전산망의 사용은 극히 제한된 범위로 축소될 수 밖에 없다. 따라서, 초고속 정보통신망의 효용극대화를 위하여 시스템 보안, 전산망 보안 및 암호와 기법 등 정보 보호 기반 기술의 꾸준한 연구 개발이 필수적이다.

정보 보호 기술의 표준화 활동은 현재 ISO, ISO/IEC JTC1, ITU-TS 등의 국제 표준화 기구와 IEEE, ECMA, ETSI, ANSI 등 지역 및 국가수준의 표준화 기구에 의해서 이루어지고 있다. 이러한 각 표준화 활동의 목적은 첫째로 암호기술을 이용한 정보 보호 방법과 보호 서비스를 표준화 함으로써 정보화 사회에서 요구되는 통신비밀의 보장 및 사생활 보호등과 같은 안전한 정보통신 서비스를 가능하게 하고, 둘째로 표준에 기초한 보호장치 및 소프트웨어의 양산을 통하여 정보 보호에 요구되는 비용경감을 가능하게 하기 위한 것이다. 특히, 미국의 경우는 전산망 보안문제에 효과적으로 대응하기 위해 미국방성의 후원으로 CERT (Computer Emergency Response Team)을 운

* 대전대학교 컴퓨터공학과 교수
** 한국전자통신연구소 책임연구원
+ 본 연구는 1995년도 한국통신 출연과제 지원에 의하여 수행된 것임.

용하고 있다. 또한 국가 컴퓨터 보안국(NCSC : National Computer Security Center)의 주관으로 TPEP(Trusted Products Evaluation Program)을 운영하고 있으며 명문화된 전문기관에 의하여 포괄적으로 분야별 심도있는 연구가 이루어지고 있다.

X.500 디렉토리 표준은 전세계적인 규모의 정보통신망을 상호연결하는데 있어서 다목적 분산 디렉토리 서비스를 구축하기 위한 기반 기술을 제공한다. 디렉토리는 사람들 또는 통신네트워크 및 기타 시스템에게 하나의 정보 근원지로서 작용하며, 여러 형태의 기관 또는 개인에 의하여 다양한 형태의 정보 서비스가 가능하다. 그러나, 정보통신망 보안 측면에서 디렉토리는 두가지 중요한 문제를 갖고 있다. 첫째는 디렉토리 정보의 기밀성 및 무결성에 대한 요구가 각 사용자별로 광범위하게 다른 정보 보호 서비스를 요구한다. 둘째는 디렉토리에 의하여 제공되는 서비스들이 다른 적용 업무들의 정보 보호 기술과 협동되어야 한다. 또한, 디렉토리 표준은 매우 방대한 내용이면서 데이타베이스 구축 기술, 분산처리기술, 정보 보호 및 통신 프로토콜 기술등이 광범위하게 복합적으로 적용된 대형 분산 시스템 개발의 중요한 이정표 역할을 한다. 디렉토리는 성격상 혼자 사용하는 것이 아니고 그 디렉토리 영역에 속하는 모든 사용자가 공통으로 사용하는 데이타베이스의 성격을 가진다. 이러한 특성 때문에 디렉토리의 내용에 불법적으로 접근하는 것을 방지하는 장치가 있어야 한다. 따라서 디렉토리 서비스를 필요로 하는 적용 업무에 있어서 통신정보 및 시스템에 대한 액세스 제어, 인증, 부인봉쇄, 데이타 기밀성, 무결성, 키관리 등의 서비스가 지원되어야 한다[3]~[5].

디렉토리는 ITU와 ISO/IEC/JTC1/SC21 협동 프로젝트로 1985년부터 개발에 착수하여 1988년도에 X.500 시리즈 권고사항으로 발표된 이래 각 국가별로도 표준화 단체 및 관련 기관에 의해 꾸준한 표준화 작업이 진행되고 있다. 반면 국내에서는 디렉토리 '88 표준을 번역, 분석한 기본 표준과 기능 표준을 작성하여 국가 표준으로 제정한바 있으나, 정보 보호 기술과 관련된 표준화 연구는 미진한 상태이다. 그 동안 국내의 경우 일반적인 정보 보호 기술에 대한 연구는 일부 관련기관에 의해 제한적으로 수행되어 선진국에 비해 극히 저조한 실정이었다. 그러나, 80년대부터 일부 통신장비의 안전성 문제에 대해 관심을 갖고 암호 알고리즘 및 이의 실용화 기술 개발 연구가 특정 기관에 의해 수행되어 왔다. 한편 정부 주도로 5대 기간 전산망 사업등과 같은 대형 프로젝트를 통하여 정보통신망이 구축 보급되면서 정보호의 필요성에 대한 인식이 새로워지고 있으나 그에 상응하는 연구개발에 대한 투자는 상대적으로 저조한 실정이다.

1992년까지 많은 기관들이 X.500 디렉토리 서버를 설치하여 약 300,000개의 엔트리를 갖고 인터네트에 연결된 370여 기관에 서비스가 이루어졌다[4]. 이제 디렉토리 표준은 일반 대규모 회사, 상업적 공중 통신 반송업자 및 전자 우편찾기와 같은 전체적 네트워크 공동체 등을 포함하여 사용환경이 방대한 중요한 사항이다. 따라서, 본 연구에서는 X.500 디렉토리 표준안의 기초적 일반 사항과 인증 및 액세스 제어를 통한 정보 보호 서비스를 살펴보고자 한다.

2. 디렉토리 모델

광범위한 정보처리 시스템의 상호 연결을 쉽게 하기 위하여 디렉토리 서비스가 제공된다. 각 정보처리 시스템은 다수의 서로 다른 제작자와 관리 체계에 의하여 구성되어 있다. 이러한 다양성은 디렉토리 시스템 구축에 있어서 여러가지 기술적 동의와 표준화의 복잡

성과 어려움을 수반하게 된다. 따라서 표준안은 그만큼 방대하면서 복잡한 내용을 담고 있기 때문에 여기서는 이를 간략히 하여 4가지 측면에서 살펴보고자 한다.

첫째로 광범위한 지역의 디렉토리 정보가 기능적으로 어떻게 분산되어 구성될 수 있고, 이들 사이의 통신을 통한 정보의 교환 및 관리 체계 측면을 기능 모델에서 다룬다. 둘째로 정보모델에서 실세계에 관심있는 정보들이 디렉토리에 논리적으로 어떻게 표현되었고, 셋째로 이러한 디렉토리 정보들이 사용자에게 어떻게 서비스되는가를 서비스 모델에서 다루고, 넷째로 디렉토리 정보 보호 모델 측면을 간략히 설명한다.

2.1 기능 모델

디렉토리는 사람, 조직, 서비스 등과 같은 실세계의 객체(Object)에 관한 정보를 보유하고 사용자에게 이들 정보에 대한 서비스를 제공하는 온라인 분산 시스템이다. 이것은 하나의 시스템에서 디렉토리 정보를 갖고 있고 디렉토리 프로토콜을 실행하는 일련의 협동하는 시스템들로 구성된다. 각 시스템에서 디렉토리의 역할을 나타내는 응용 프로세스를 DSA (Directory System Agent)라고 하며, 디렉토리는 이 DSA 통신 집합으로 모델된다[6][7][8].

디렉토리의 사용자는 일종의 client로써 사람이나 컴퓨터 프로그램들인 반면에 server로 볼 수 있는 디렉토리는 사용자측에게 디렉토리 정보에 대한 조회(retrieve)와 수정(modify) 2 가지 범주의 기본 서비스를 제공한다. 사용자는 DUA (Directory User Agent)라고 하는 응용 프로세스에 의하여 디렉토리와 상호작용을 하는데 (그림 1)은 이러한 디렉토리와 사용자 모델을 나타내고 있다[4][5][6].

사용자는 디렉토리의 서비스를 제공받기 위하여 DUA를 통하여 접근할 수 있도록 되어 있으며 각 DUA는 정확히 하나의 디렉토리 사용자를 나타낸다.

하나의 DUA와 DSA 또는 2개의 DSA가 상호 디렉토리 서비스를 원하는 특정 쌍의 응용 프로세서가 다른 시스템에 위치될 수 있다. 이런 경우는 다음과 같은 OSI 디렉토리 프로토콜에 의하여 수행된다.

- DAP(Directory Access Protocol) : 디렉토리 서비스를 요구하는 하나의 DUA와 DSA 사이의 프로토콜
- DSP(Directory System Protocol) : 디렉토리 서비스의 연결을 지원하는 두 DSA 사이의 프로토콜

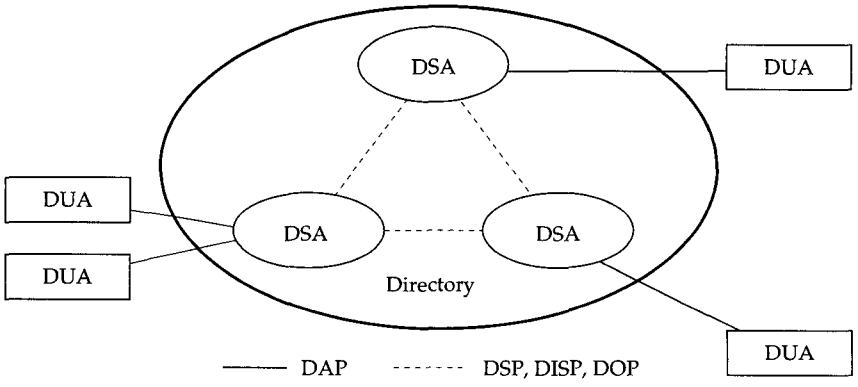


그림 1 디렉토리와 사용자 모델

- DISP(Directory Information Shadowing Protocol) : 복제 정보의 교환을 지원하기 위하여 shandowing 등의 선정에 관한 두 DSA 사이의 프로토콜
- DOP(Directory Operational Binding Protocol) : 두 DSA 사이의 관리 정보 교환에 관한 프로토콜

DUA로 부터 디렉토리 서비스 요청을 받은 DSA는 지역내에서 접근 가능한 정보일 때는 직접적으로 서비스할 수 있지만, 그렇지 않을 경우는 다른 DSA로 요청을 넘겨주는 chaining, 다른 DSA를 접촉할 수 있도록 소개하는 referal, 또는 다수의 DSA에게 동시에 요청을 넘겨주는 multicasting 기법이 사용된다. DUA가 특정의 DSA에 연결될 필요는 없으며, 주어진 요청을 처리하기 위하여 어느 DSA를 사용할 것인지 자신의 결정을 따를 수 있다. 디렉토리의 객체 정보는 서비스 수행의 편의를 위하여 다수의 DSA에 복제되는 것이 가능한데 이러한 복제의 형태를 다루는 표준절차를 shadowing이라 한다.

한편, 몇 개의 DSA와 DUA들의 집합이 하나의 구분된 기관에 의해서 운영될 때 이것을 DMD(Directory Management Domain)이라고 한다. 그리고 이 DMD를 운용하는 기관이 공중통신 서비스를 제공하는 공공기관일 때는 ADDMD(Administration DMD)이고, 사설인 경우는 PRDMD(Private DMD)이라고 한다. DMD에 의한 DUA의 관리는 보수유지 및 어떤 경우의 소유 권한과 같은 서비스에 대한 책임 수행을 의미한다. 즉, 이름관리, 서브 스키마 관리, 정보 보호 관리 등과 같은 디렉토리의 어떤 관리적 측면에 책임이 있는 관리 기관들이 있으며, 특히 정보 보호의 측면을 구체적으로 다루는 관리 기관을 Security Authority라고 한다.

2.2 정보 모델

디렉토리 사용자가 관심이 있는 객체들에 관한 전체적 정보는 DIB(Directory Information Base)라고 하는 논리적 디렉토리로 구성되며 이 정보의 저장은 많은 물리적 시스템들 사이에 분산될 수 있다. DIB는 국가, 조직, 사람 등과 같은 객체에 관한 정보가 들어있는 엔트리(entry)들로 구성되어 있다. DIB의 엔트리들은 트리의 형태로 배열되어 있는데 이것을 DIT(Directory Information Tree)라고 한다. 이러한 엔트리들의 배열은 객체들사이의 자연적인 계층관계를 반영한 것이다. (그림 2)는 DIT의 개념적 구조를 나타내고 있다[6][9].

DIT는 엔트리들의 트리 구조로 되어 있으며, 이 엔트리는 속성(attribute)들의 집합으로 구조화되어 있다. 디렉토리에 의해서 보유된 정보들이란 이 엔트리들의 각 속성들로서 저장되어 있는 것이다. 속성은 속성타입(attribute type)과 그 타입에 해당하는 하나 또는 그 이상의 속성값(attribute value)들로 구성되어 있다.

디렉토리에 표현된 각 객체는 이름에 의하여 식별된다. 따라서 2개의 객체가 같은 이름을 갖을 수 없으며 디렉토리에 있는 객체의 이름들은 애매모호해서는 안된다. 그러나, 그 이름이 유일한 하나를 의미하는 것은 아니며, 단지 객체를 명확하게 표현할 필요가 있음을 의미한다. 이름에는 DIT의 루트로부터 해당 엔트리까지 전체경로명을 나타내는 고유이름(Distingushed Name)과 DIT의 각 계층별로 존재하는 엔트리에게 부여된 상대이름(Relative Distingushed Name)이 있다[6]. 또한, 한 객체에 대한 원래의 이름외에 다른 이름 경로를 통하여 접근할 수 있도록 하는 가명(alias name)이 별도의 엔트리를 차지하게 할 수도 있다. 이외에 의미 이름(purported name)은 사용자가 디렉토리에서 어떤 서비스를 요청할 때 부여한 이름으로써 아직 디렉토리 시스템

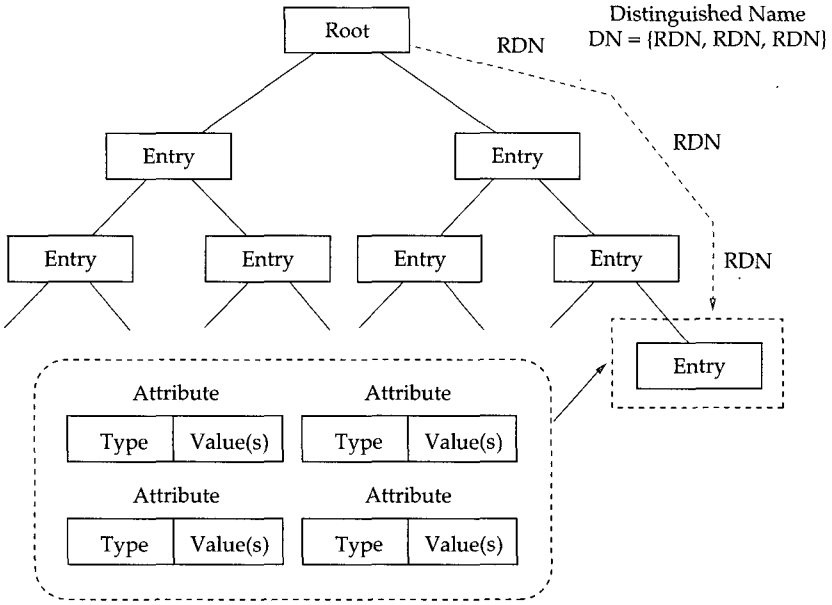


그림 2 DIT의 개념적 구조

에 의해 유효한 이름으로 채택되지 않았음을 나타내기 위하여 구분되어 사용된다.

2.3 서비스 모델

디렉토리는 사용자에게 여러가지 서비스 제공 능력을 가진 하나의 통합된 시스템으로 보여진다. 이러한 서비스들은 디렉토리에게 정보를 질의하고 또한 디렉토리 정보를 수정할 수 있도록 사용자에게 허용한다. 사용자의 DUA가 디렉토리로 서비스로 요청하면 디렉토리는 요청을 처리하여 결과를 DUA에게 돌려주도록 한다. 디렉토리의 서비스는 (그림 3)에 나타난 것과 같이 사용자와 디렉토리 사이에 DUA 및 접근점(access point)을 통해서 제공된다.

디렉토리의 접근점에는 몇 가지 서로 다른 서비스의 조합을 나타내는 어떤 타입들이 있는데 이것은 일종의 포트(port)로 볼 수 있으며 디렉토리는 이러한 포트를 제공하는 하나의 객체(object)로 생각할 수 있다. 각 포트는 디렉토리가 DUA와 함께 작용할 수 있는 특별한 종류의 상호 관계를 정의한다. 즉, 디렉토

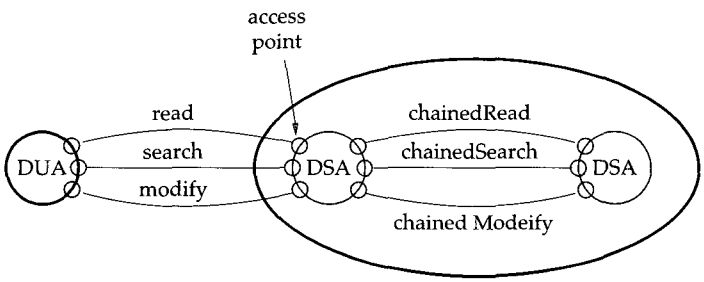


그림 3 분산 디렉토리 서비스 모델

```
directory    OBJECT
             PORTS
             {  readPort      [S],
                searchPort    [S],
                modifyPort    [S]}
: : = id-ot-directory
```

(a) 디렉토리 객체

```
dua    OBJECT
       PORTS
       {  readPort      [C],
          searchPort    [C],
          modifyPort    [C]}
: : = id-ot-dua
```

(b) DUA 객체

```
readPort    PORT
            CONSUMER INVOKES { Read, Compare, Abandon }
: : = id-pt-read
searchPort    PORT
              CONSUMER INVOKES { List, Search }
: : = id-pt-search
modifyPort    PORT
              CONSUMER INVOKES { AddEntry, RemoveEntry, ModifyEntry, ModifyRDN }
: : = id-pt-modify
```

(c) 디렉토리 오퍼레이션

그림 4 디렉토리 서비스 모델

리는 DIB에 있는 특정 엔트리로 부터 정보를 읽을 수 있도록 하는 Read Port, DIB의 탐색을 허용하는 Search Port, 그리고 DIB의 엔트리를 수정할 수 있도록 하는 Modify Port 3가지의 오퍼레이션을 제공한다. DUA는 디렉토리가 제공하는 서비스를 소비하는 것으로써 마찬가지 방법으로 정의할 수 있다. (그림 4)는 서비스 측면에서 본 디렉토리와 DUA에 대한 정의를 나타내고 있다[6].

- Read : 특정 엔트리로부터 원하는 속성값을 읽음
- Compare : 명시된 엔트리의 속성값을 사용자가 제공하는 값과 비교
- Abandon : 어떤 질의 서비스 요청에 대해서 수행되기 전에 취소
- List : DIT에 지정된 엔트리의 모든 하위 엔트리들을 출력
- Search : 제시된 조건에 맞는 DIT의 지정된 범위내의 엔트리로 부터 정보를 제공
- AddEntry : DIT에 새로운 엔트리를 추가
- RemoveEntry : DIT에서 지정된 엔트리를 삭제
- ModifyEntry : 지정된 엔트리에 대하여 추가, 삭제, 속성 타입이나 값의 변경
- ModifyRDN : 엔트리의 상대명(RDN)을 수정

Read, Compare, List 및 Search는 질의 오퍼레이션의 범주로 보고 AddEntry, RemoveEntry, ModifyEntry와 ModifyRDN은 수정 오퍼레이션으로 크게 나누어 생각할 수 있다. 디렉토리 서비스를 요청하는 이러한 오퍼레이션들은 일련의 service control를 사용하여 제어될 수 있는데, 이것은 요청된 서비스에 사용자가 원하는 처리 지침이나 제한조건을 반영하게 된다. 예를 들면, 요청처리에 관한 최대시간, 반환결과의 최대크기, 요청의 우선순위 등

이 있다. 디렉토리는 각 DUA 요청에 대하여 성공된 결과나 또는 에러를 반환함으로써 응답한다. 에러의 반환은 어떤 문제 때문에 디렉토리가 정상적으로 요청을 처리할 수 없음을 나타낸다. 즉, 존재하지 않는 엔트리의 이름을 요청했거나, 계속적인 처리를 방해하는 디렉토리의 어떤 내부적인 문제가 있거나 또는 사용자가 원하는 어떤 행위를 할 수 있도록 권한이 부여되어 있지 않는 경우이다.

2.4 정보 보호 모델

디렉토리는 수많은 사람들 또는 통신 네트워크 및 기타 시스템에게 하나의 정보 근원지로써 작용하며, 여러 형태의 기관 또는 개인에 대하여 다양한 형태의 정보 서비스가 가능하다. 그러나 정보통신망 보안 측면에서 디렉토리는 두가지 중요한 문제를 갖고 있다. 첫째는 디렉토리 정보의 기밀성(data confidentiality)과 무결성(data integrity)등에 관한 요구가 사용자별로 광범위하게 다른 디렉토리 서비스를 요구한다. 둘째는 디렉토리에 의하여 제공되는 서비스들이 다른 일반적 응용 업무들의 자체 정보 보호 기술과 협동되어야 하는 점이다. 이와 같은 문제들을 고려하여 여러가지 기관에 의한 정보 보호 정책과 일치하도록 디렉토리 정보 보호 모델이 구성되어야 한다.

디렉토리시스템에 관한 정보 보호 대책으로는 authorization과 authentication 두 가지의 정책으로 나눌 수 있다[6]. authorization은 디렉토리 사용자가 미리 정의된 액세스 제어 정책(access control policy)에 일치하는 액세스 권한을 가진 사용자인지 판별하여 규정된 액세스만을 인정하는 것이며 authentication은 DSA와 디렉토리 사용자의 신분, 그리고 접근점에 접수된 정보의 발신처 신분을 확인하는 인증 서비스를 제공하는 것이다. 액세스 제어 정책에는 디렉토리의 어떤 영역내에 있는 엔트리 또는 속성에 대한 불법적인 탐색, 시험, 수정 등의 조작을 보호하는 수단이 제공되며, 이러한 조작을 수행하기 위한 사용자의 권한에 관한 정보가 DSA에게 사용가능해야 할 것이다. 디렉토리 정보에서 보호되어야 하는 범위는 각각 DIT의 전체, 서브트리, 각 엔트리, 엔트리내의 전체 속성, 그리고 선택된 속성값들로 나누어 보호할 수 있다[6].

본 디렉토리 모델의 authentication에서는 사용자의 이름과 패스워드 등을 사용하는 단순 인증과 공개키를 이용한 암호화 기법 등을 사용하는 강력 인증 방법이 제시되고 있다[6]. DAP와 DSP는 발신자 및 요청에 대한 서명을 하고 응답자 및 결과에 대한 서명에 의하여 발신자와 응답자를 상호 인증하고 요청과 결과 정보의 무결성에 대한 강력 인증을 제공할 수 있다. 또한 DAP는 DUA와 DSA사이 그리고 DSP는 두 DSA 사이의 단순 인증을 제공할 수 있다. 접수되는 정보의 무결성은 지역적 문제이지만, 이것은 디렉토리 정보 보호 정책과 일치되어야 한다. 적용 업무에 있어서 관리기관은 지역적 디렉토리 사용자에게 자신의 정보 보호 정책을 사용하는데 디렉토리의 이용을 허용한다. 디렉토리는 통신 주체 사이의 인증에 필요한 정보(DN, 패스워드, 인증서 등)을 갖고 적용 업무를 지원할 수 있다.

3. 디렉토리 정보 보호 요구사항

정보통신망을 이용한 응용시스템에서 디렉토리는 각 응용 분야별 자신의 정보 보호 요구 조건들을 갖고 있다. 가장 기본적인 요구사항은 디렉토리에 갖고 있는 정보의 기밀성 및 무결성에 대한 것이다. 어떤 정보의 소유자는 임의의 사용자들에게 특정한 정보의 공개 범위를 제한하거나, 또는 그 정보에 대하여 수정할 수 있는 사용자를 통제하고 싶을 것이다. 그러나, 디렉토리는 그 용도의 특성상 자연스

럽게 어느 누구에게든지 액세스 시도가 이루어질 수 있는 다음과 같은 특징이 있다.

1) 저장성 : 사람, 기관, 응용서비스등 모든 객체에 대한 정보의 저장
2) 노출성 : 임의의 시간 및 장소에서의 임의의 엔티티로부터 접근될 수 있는 노출 문제

즉, 디렉토리 시스템은 수많은 분야의 다수의 사용자가 공동으로 사용하는 정보 근원지 역할을 하기 때문에 활용측면에서 광범위하고 중요하면서도 모든 사용자에게 노출된 여러가지의 위협 요소들을 갖고 있다. 따라서, 디렉토리 정보에 대한 항목별 정보 보호 서비스와 이를 달성하기 위한 정보 보호 메카니즘이 표준안에 명시되고 있다[6].

3.1 정보 보호 위협

1) 신분 가로채기(identity interception) : 정보를 오용하기 위해 통신에 개입된 것으로 파악되는 사용자 신분
2) 위장(masquerade) : 정보를 액세스하거나 부가적 권한을 얻기 위하여 다른 사용자로 위장
3) 재전송(replay) : 통신 내용을 기록하여 나중에 재전송
4) 데이타 엿듣기(data interception) : 비인가된 사용자가 통신중인 사용자 데이타를 관찰
5) 불법조작(manipulation) : 비인가된 사용자가 통신상의 데이타를 교체, 삽입, 삭제 또는 순서 조작
6) 부인(repudiation) : 통신의 일부 또는 전체에 개입한 사실을 사용자가 부정
7) 서비스 거절(denial of service) : 통신의 방해나 중단 또는 시간이 위급한 오퍼레이션의 지연
8) 불법통로 조작(mis-routing) : 의도하는 사용자로의 통신 경로에 대한 불법적 조작(이것은 인증 골격의 범주밖이지만 본 인증 골격에서 제공되는 적절한 정보 보호 서비스를 통하여 회피하는 것이 가능)
9) 트래픽 분석(traffice analysis) : 사용자들 사이의 통신에 관한 정보 감시(어떤 특정 OSI layer에 제한되어 있지 않는 본 인증 골격 범주 밖이지만 적절한 암호화 또는 랜덤 데이타를 사용하여 부가적인 이해할 수 없는 트래픽을 생성하여 끼워 넣으므로써 부분적 보호 가능)

3.2 정보 보호 서비스

위협을 방지하기 위하여 여러가지 정보 보호 서비스가 제공될 필요가 있다. 인증 골격에 의해 제공되는 보호 서비스들은 다음과 같다.

1) 대등 실체 인증(peer entity authentication) : 통신의 어느 순간에 있는 사용자의 신분을 확인하는 서비스로써 사용자 신분에 관한 위장 및 재전송을 방지하기 위해 사용된다. 그리고 두 가지의 다른 대등 실체 확인 서비스가 다음과 같이 요청될 수 있다.
   - 자료 발신 또는 수신 실체 인증(single entity authentication)
   - 통신상대인 두 사용자간의 상호 인증(mutual authentication)
2) 데이타 기밀성(data confidentiality) : 비인가된 노출로부터 자료의 보호를 위해 사용되는 서비스로써 자료의 엿듣기 방지를 위해 사용될 수 있다.
3) 데이타 무결성(data integrity) : 통신상의 데이타 무결성을 보증하기 위한 서비

스로써 불법적 조작을 탐지하거나 또는 방지를 위해 사용될 수 있다.

4) 부인봉쇄(non-repudiation) : 데이타의 무결성 및 발신처를 증명하고 third party에 의하여 어떤때든지 확인할 수 있도록 쌍방이 위조할 수 없는 관계를 증명하는 서비스

이상과 같은 정보 보호 서비스를 수행하는 정보 보호 메카니즘은 다음과 같다.

1) 인증 교환(authentication exchange) : 수신자에 의하여 검사될 수 있는 이름과 패스워드를 기초로 한 단순 인증과 비대칭 암호 기법을 사용하는 강력 인증 방법이 있다. 이 인증교환 방식은 대등실체 인증 서비스를 지원하기 위하여 사용된다.
2) 암호화(encipherment) : 전송중에 대칭 또는 비대칭 구조를 사용한 자료의 암호화 기법을 사용하는 것으로써 데이타 기밀성 서비스를 지원한다.
3) 디지탈 서명(digital signature) : 이 방법은 전달될 관련자료의 압축된 스트링을 발신자와 비밀키로 암호화 하며 순수 데이타의 함께 디지탈 서명이 수신자에게 보내진다. 이것은 비대칭 암호 기법이 사용되며 데이타의 무결성 서비스와 부인 봉쇄 서비스를 지원한다.

(그림 5)은 X.509 디렉토리 인증 골격에서 규정된 정보 보호 위협과 이를 방지하기 위한 정보 보호 서비스 및 메카니즘을 요약하여 나타내고 있다.

4. 디렉토리 인증 서비스

X.509는 디렉토리 사용자에게 인증 서비스의 규정에 대한 골격을 정의한다. 디렉토리는 공개키 인증서의 저장소로서 사용될 수 있으며, 각 인증서는 사용자의 공개키를 갖고 있는데 신뢰되는 CA(Certification Authority)의 개인키로 서명되어 있다. X.509는 X.500 디렉토리 서비스가 널리 사용될 것으로 기대되기 때문에 중요한 표준안으로써 X.509에 정의된 인증 구조와 인증 프로토콜은 다른 배경에서도

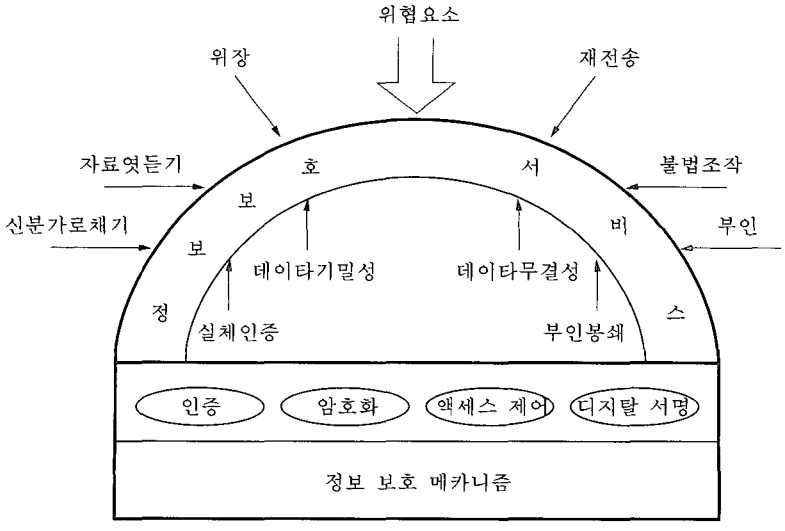


그림 5 디렉토리 위협 요소 및 정보 보호 서비스

다양하게 응용될 수 있다. X.509는 공개키 암호 및 디지탈 서명의 사용에 기초하고 있으며, 이 표준은 특정 알고리즘의 사용을 묘사하지 않지만 RSA를 권고하고 있다[6].

4.1 인증서 형식

X.509 스키마의 핵심은 각 사용자에게 연관된 공개키 인증서이다. 이 사용자 인증서들은 어떤 신뢰할 수 있는 CA에 의하여 생성되어 CA 또는 사용자가 디렉토리에 배치하는 것으로 가정된다. 디렉토리 서버 자체는 공개키의 생성이나 또는 인증 기능에 대한 책임이 없으며, 단지 사용자에게 인증서를 획득하는데 쉽게 접근할 수 있는 장소를 제공한다. (그림 6)은 인증서의 일반적 형태를 보여주고 있다.

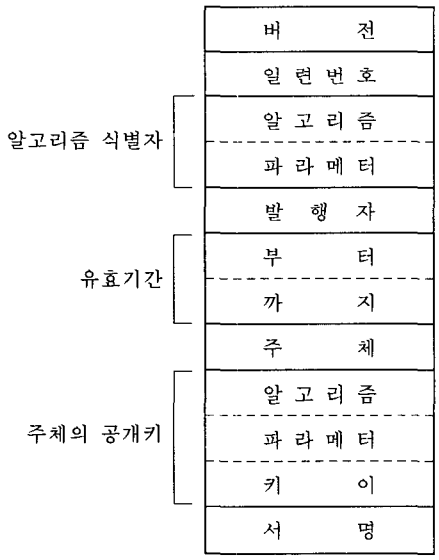


그림 6 X.509 인증서

- 버전(V) : 인증서의 계속되는 버전을 구별하기 위한 것으로 기본이 1988로 되어 있다.
- 일련번호(SN) : 본 인증서와 명확히 연관된 CA내의 유일한 정수값.
- 알고리즘 식별자(AI) : 인증서에 서명하기 위해 사용된 알고리즘 및 연관된 파라메터.
- 발행자(CA) : 본 인증서를 생성하고 서명한 CA.
- 유효기간(T_A) : 인증서가 유효한 시작과 끝 두 날짜
- 주체(A) : 본 인증서가 참조하는 사용자
- 공개키 정보(A_P) : 주체의 공개키 및 이 키가 사용되는 알고리즘의 식별자
- 서명 : 인증서의 기타 모든 부분들로써 CA의 개인키로 암호화된 기타 부분들의 해쉬 코드로 구성된다.

인증서를 정의하기 위하여 다음 표기법이 사용된다.

$$CA \ll A \gg = CA\{V, SN, AI, CA, T_A, A, A_P\}$$

여기서, $Y \ll X \gg$ 는 인증기관 Y에 의해서 발행된 사용자 X의 인증서이며, $Y\{I\}$는 Y에 의한 I의 서명으로써 I에 암호화된 해쉬 코드가 부가되어 구성된다.

CA는 자신의 비밀키로 인증서를 서명하는데, 만일 대응하는 공개키가 사용자에게 알려져 있다면, 그때 사용자는 CA에 의하여 서명된 인증서가 유효한지 확인할 수 있다.

4.2 인증서 획득

CA에 의하여 생성된 사용자 인증서는 다음과 같은 특성을 갖는다.

- CA의 공개키를 액세스하는 어떤 사용자는 인증되었던 사용자 공개키를 복구할 수 있다.
- CA이외의 어느 누구도 검출되지 않고 인증서를 수정할 수 없다.(위조 불가)

인증서는 위조할 수 없기 때문에 그것을 보호하기 위한 특별한 노력없이 디렉토리에 배치될 수 있다. 만일 모든 사용자가 동일한 CA에 가입한다면 그때 그 CA의 공통된 신뢰하에 있게 된다. 사용자 인증서는 모든 사용자에게 의하여 액세스되도록 디렉토리에 배치될 수 있고, 또한, 한 사용자가 인증서를 다른 사용자에게 전송할 수 있다. 어느 경우든 일단 B가 A의 인증서를 소유하고 있으면, A의 공개키를 갖고 암호화된 메시지가 도용으로부터 안전하고 A의 개인키를 갖고 서명된 메시지가 위조할 수 없다는 확신을 갖게 된다.

만일 사용자 공동체가 매우 크다면, 모든 사용자를 하나의 동일한 CA에 가입시키는 것이 현실적이지 못할 수 있다. 인증서에 서명을 하는 것이 CA이기 때문에 각 가입된 사용자는 서명을 확인하기 위하여 CA 자신의 공개키 복사물을 갖고 있어야 한다. 이 공개키는 사용자가 연관된 인증서에 확신을 갖도록 절대적 안전한 방법으로 각 사용자에게 제공되어야 한다. 따라서 많은 사용자들이 수많은 CA들에 각각 가입되는 것이 보다 현실적일 수 있으며, 각 CA는 사용자 그룹들에게 자신의 공개키를 보다 안전하게 제공할 수 있게 된다. 예를 들면 A가 CA X_1으로부터 인증서를 획득하고 B가 CA X_2로부터 인증서를 획득했다고 가정하자. 만일 A가 X_2의 공개키를 안전하게 알지 못한다면, 그 X_2에 의해 발행되는 B의 인증서는 A에게 소용이 없다. A는 B의 인증서를 읽을 수 있지만, A는 서명을 확인할 수 없다. 그러나, 만일 두 CA가 안전하게 그들 자신의 공개키를 교환한다면 다음의 절차가 A로 하여금 B의 공개키를 획득할 수 있도록 할 것이다.

1) A가 X_1에 의해 서명된 X_2의 인증서를 디렉토리로부터 획득한다. A는 안전하게 X_1의 공개키를 알기때문에 인증서로부터 X_2의 공개키를 획득할 수 있고, 인증서에 있는 X_1의 서명을 사용하여 그것을 확인할 수 있다.
2) A는 그때 디렉토리로 되돌아가서 X_2에 의해 서명된 B의 인증서를 획득할 수 있다. A는 이제 X_2의 신뢰할 수 있는 복제된 공개키를 갖고 있기 때문에 A는 서명을 확인할 수 있고 B의 공개키를 안전하게 획득할 수 있다.

A는 B의 공개키를 획득하기 위하여 다음과 같은 인증서의 체인을 사용한다.

$$X_1 \ll X_2 \gg X_2 \ll B \gg$$

동일한 방법으로 B는 역방향 체인을 이용하여 A의 공개키를 획득한다.

$$X_2 \ll X_1 \gg X_1 \ll A \gg$$

이러한 체인은 두개의 인증서에 제한되는 것이 아니고 다음과 같이 N개 요소로 구성된 체인을 형성할 수 있다.

$$X_1 \ll X_2 \gg X_2 \ll X_3 \gg \ldots X_N \ll B \gg$$

이 경우에 체인(X_i, X_{i+1})의 각 CA쌍은 서로가 인증서를 갖고 있어야 한다. 이와 같이 CA에 의한 모든 CA의 인증서들은 디렉토리에 표현될 필요가 있으며 사용자는 각 인증서들이 다른 사용자의 공개키 인증서 경로를 따라서 어떻게 연결되어 있는지를 알 필요가 있다. X.509는 진행과정이 직선적으로 이루어지도록 CA를 계층적으로 정렬할 것을 제시하고 있다. (그림 7)은 그러한 계층적 구조의 예를 나타내고 있다. 연결된 원은 CA들 사이의 계층적 관계를 나타내며, 연관된 네모 상자들은 각 CA 엔트리에 대하여 디렉토리에 유지되고 있

는 인증서들을 나타낸다. CA X에 대한 디렉토리 엔트리는 2가지 타입의 인증서를 포함하고 있다.

- forward certificate : 다른 CA에 의해 생성된 X의 인증서
- reverse certificate : X가 생성한 다른 CA의 인증서

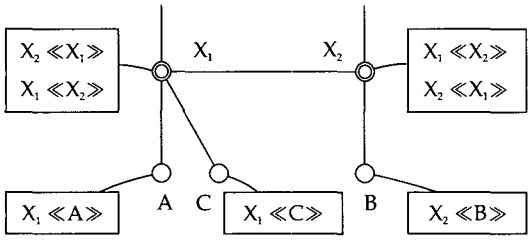


그림 7 X.509 CA 계층 구조의 가상 예

4.3 인증서 취소

인증서 형식을 보면 각 인증서는 마치 신용카드처럼 유효기간을 포함하고 있다. 일반적으로 새로운 인증서가 구인증서의 기간이 완료되기 직전에 발행된다. 그러나, 다음과 같은 이유에서 만료기간 이전에 인증서를 취소할 경우가 있을 수 있다.

1) 사용자의 비밀키가 위험하다고 생각된다.
2) 사용자가 해당 CA로 부터 더 이상 인증되지 않는다.
3) CA의 비밀키가 위험하다고 생각된다.

각 CA는 기한 만료된 인증서는 제외하고 사용자 및 다른 CA에게 발행된 인증서를 포함하며 취소된 모든 인증서들의 리스트를 유지해야 한다. 이러한 리스트들은 또한 디렉토리에 배치되어야 한다. 디렉토리에 배치된 각 인증서 취소 리스트는 발행자에 의하여 서명되고 발행자의 이름, 리스트가 생성된 날짜, 및 각 취소된 인증서에 대한 엔트리로 구성된다. 각 엔트리는 인증서의 일련번호와 그 인증서에 대한 취소 날짜로 구성된다.

사용자가 인증서를 메시지에 받을때, 그 사용자는 인증서가 취소된 것인지 결정해야 한다. 사용자는 인증서를 받을 때마다 디렉토리를 검사할 수 있으며, 디렉토리 탐색과 연관된 시간 지연을 피하기 위하여 사용자는 인증서들의 국부적 cache와 취소된 인증서의 리스트를 유지할 수 있다. 한편, 기한 만료된 인증서는 보통 디렉토리로부터 삭제되겠지만, 이것은 응용분야의 정보보호정책으로써, 만일 부인 봉쇄 서비스를 제공하고 싶다면 일정기간동안 구인증서를 유지할 책임이 CA에게 있다.

4.4 인증 교환

X.509는 다양한 적용을 위하여 3가지의 인증 절차를 포함하고 있다. 3가지는 모두 인증정보의 교환되는 수효가 다른데 결과적으로 관계자간의 다른 보증 유형을 제공한다. 이 방법들 모두는 공개키 서명을 이용하여 상대방의 공개키를 서로 알고 있다고 가정한다. 이것은 디렉토리로부터 서로의 인증서를 획득하거나 인증서가 양쪽으로부터 초기 메시지에 포함되기 때문에 가능하다. (그림 8)은 X.509의 강력 인증 절차의 3가지 방법을 나타내고 있다.

(1) 단방향 인증(One-Way Authentication)

단방향 인증은 DN에 발신측 개인키로 서명된 인증 토큰을 추가하여 송신함으로써 성취되며 다음과 같은 서비스를 제공한다.

1) 실제로 메시지가 A에 의해서 생성되었다는 A의 신분 확인

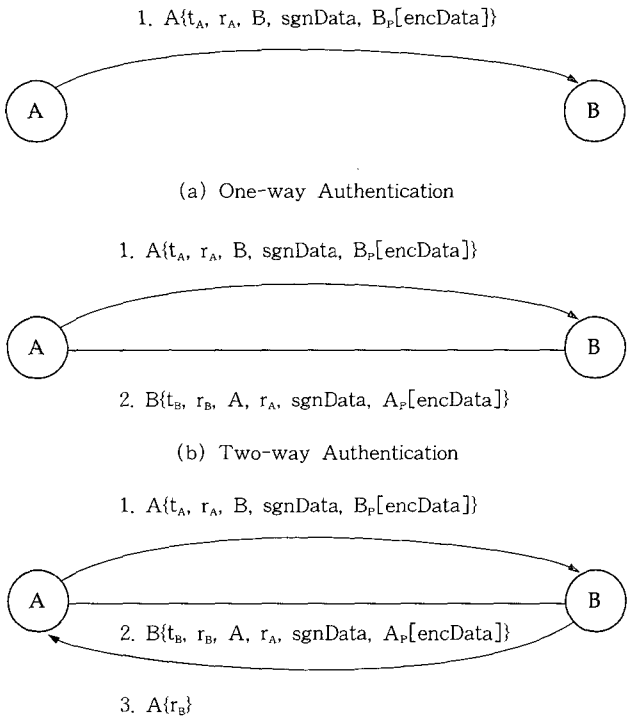


그림 8 X.509 강력 인증 절차

2) 메시지가 B측으로 보내질 의도라는 것
3) 메시지의 무결성 및 다수로 송신되지 않았다는 고유성

이 경우는 단지 발신처의 신분만 확일될 뿐이며 응답측은 아니다. 포함되는 메시지는 타임 스탬프 t_A, 고유번호 r_A, B의 신분 및 A의 공개키로 서명된 것을 포함한다. t_A는 토큰의 생성 및 만료 날짜로 구성되며 이것은 메시지의 지연 배달을 방지한다. 고유번호 r_A는 재전송 공격을 방지하기 위해 사용되며 이 번호는 메시지 만료되기까지의 범위내에서 유일해야 한다. 따라서 B는 기간이 만료될 때까지 그 번호를 저장하고 동일한 번호를 갖는 새로운 메시지는 거부할 수 있다. 순수한 의미의 인증은 A가 B로 보내는 신뢰성 확인을 위하여 단순히 사용되는 것이지만 메시지에는 또한 전송될 정보들을 포함할 수도 있다. 이러한 정보 sgnData가 그것의 인증성과 무결성을 보장하면서 서명의 범주내에 포함된다. 이 메시지는 또한 B의 공개키로 암호화된 session key를 B로 전송하기 위해서 사용될 수도 있다.

(2) 양방향 인증(Two-Way Authentication)

단방향 인증에서 제공되는 3가지 요소외에 다음의 사항들이 추가된다.

4) 응답으로 생성된 메시지가 실제로 B에 의하여 생성되었다는 B에 대한 신분 확인
5) 메시지가 A측으로 보내질 의도라는 것
6) 응답의 무결성 및 고유성

양방향 인증은 통신상의 양쪽측이 서로 상대방의 신분 확인을 허용한다. 응답 메시지는 응답을 확인하기 위하여 A로부터의 고유번호를 포함하며, 또한 B에 의하여 생성된 time-stamp와 고유번호, 그리고 서명된 추가정보 및 A의 공개키로 암호화된 session key를 포함할 수 있다.

(3) 세방향 인증(Three-Way Authentication)

세방향 인증에서는 양방향 인증과 동일한 성질을 갖고 있지만 고유번호 r_B의 서명된 복제를 갖고 있는 A로부터 B로의 마지막 메시지가 포함된다. 이러한 설계의 목적은 time-stamp가 검사될 필요가 없다는 것이다. 즉, 각 고유번호가 양측에 반향되기 때문에 재 전송 공격을 검출하기 위하여 반향되는 고유번호를 검사할 수 있다.

4.5 인증서 및 키 관리

인증서의 생산은 오프라인으로 실행되어야 하며, 자동화된 조회/응답 방식으로 해서는 안된다[6]. 이러한 인증서의 장점은 CA의 비밀키가 격리되고 물리적으로 안전하여 결코 알려질 수 없으며 단지 CA 자신의 공격에 의해서만 습득될 수 있다는 것이다. 인증서는 정보의 공개적 사용 가능한 부분인데, 디렉토리로 전송을 위하여 특별히 필요한 보안 수단은 없다. 생성된 인증서가 주어지면 사용자는 단지 디렉토리 액세스를 통하여 엔트리에 저장할 뿐이다. CA는 인증서에 대하여 다음과 같은 관리가 이루어져야 한다.

- CA는 인증서를 발행하기 이전에 사용자의 신분에 만족되어야 하며 두 사용자에게 동일한 이름으로 인증서를 발행해서는 안된다.
- 인증서는 유효기간을 갖고 있어야 하며, CA는 서비스의 계속성을 위하여 적기에 대체 인증서의 사용성을 보장해야 한다.
- 사용자 또는 CA의 비밀키가 위험하거나 또는 CA에 의해 더 이상 사용자가 인증되지 않을 때 인증서 취소에 대한 관리가 있어야 한다. CA는 자신이 발행하여 취소된 인증서들의 time-stamp 리스트를 유지해야 한다. 그리고, 모든 CA 들의 취소된 인증서 time-stamp 리스트가 대상 CA에게 알려져야 한다.

구현을 하기 위한 전체적 정보 보호 정책은 키 쌍의 생명주기를 정의하겠지만 이것은 X.509 인증 골격의 범주밖이다. 그러나 모든 비밀키가 해당되는 사용자에게만 알려져야 함은 매우 중요한 문제다. 키이는 사람들이 기억하기 쉽지 않기 때문에 그것을 기억하기 위한 적절한 방법으로 smart card를 사용할 수 있다. 즉, 이것은 비밀키, 그리고 선택적으로 사용자의 공개키, 사용자 인증서 및 인증기관의 공개키 사본을 보관시킬 수 있다. 또한, 이 카드 사용의 안전성을 위하여 카드소유자에게 별도의 개인 식별 번호(Personal Identification Number)을 요구함으로써 보호 효과를 높일 수 있다.

사용자의 키 쌍을 생성하는데는 다음의 3가지 방법이 있다.

1) 사용자가 자신의 키 쌍를 직접 생산하는 방법으로써 이것은 사용자의 비밀키가 다른 엔티티에 결코 공개되지 않는다는 장점이 있다.
2) 제 3자가 키 쌍을 생성하는 방법으로써 제 3자는 실질적으로 안전한 방법에 의하여 사용자에게 비밀키를 알릴 수 있도록 강구되어야 한다.
3) 키 쌍이 CA에 의하여 생성되는 방법으

로써 제 3자 생성하는 방법의 특별한 경우로 볼 수 있다. 이 방법은 CA가 이미 사용자에게 신뢰되어 있고 필요한 물리적 보호 수단이 제공되므로 별도의 안전한 자료전달을 요구하지 않는 장점이 있다[6].

5. 디렉토리 액세스 제어 서비스

X.509 디렉토리 표준에서 인증 골격 이외의 정보보호에 관한 또 다른 하나의 주제는 디렉토리에 있는 정보에 대한 액세스 제어에 관한 사항이다. 액세스 제어 스킴에는 기본 액세스 제어(Basic Access Control) 및 단순화된 액세스 제어(Simplified Access Control)가 있다. 제어할 정보대상은 디렉토리의 사용자 정보, DIT 구조와 관련된 디렉토리 정보, 그리고 ACI(Access Control Information)정보를 포함하는 디렉토리 운용 정보로 나눌 수 있다. 이러한 정보에 대한 비인가된 탐지, 공개, 수정에 대한 방어 대책이 필요하다. 이 액세스 제어 구조는 일종의 액세스 제어 리스트 메카니즘을 사용하는데, 이것은 액세스 제어 대상, 액세스 허가 범주, 액세스 제어 순위, 액세스 제어 영역 등의 디렉토리에 대한 액세스 제어 정보(ACI)들로 구성된다. ACI는 일종의 액세스 제어 명령문의 집합으로 볼 수 있으며, 특정한 엔트리 또는 특정한 관리 영역내에서의 제어권을 설정한다[6].

5.1 액세스 제어 대상

액세스 제어 대상은 DIT의 엔트리 또는 속성 등의 보호 대상(protected item)과 디렉토리의 사용자들에 대한 사용자 클랙스(user class)로 나누어 허가할 수 있다. 두가지 모두 동일한 정보를 포착하지만, 구조적인 편의성 및 표현상의 효율성에 대한 차이가 있다.

protected item 명세는 엔트리 및 속성 레벨의 서브명세로 구분할 수 있다. 엔트리의 존재 자체를 감추고 싶거나 DIT의 새로운 위치로 엔트리를 옮기는 것은 엔트리 레벨에서 제어를 하고, 특정 엔트리의 어떤 속성에 대한 액세스는 속성레벨에서 제어할 필요가 있다. 속성 레벨 명세는 다음과 같은 항목들이 선택될 수 있다.

- all-user-attribute-type : 속성값을 제외한 모든 사용자의 속성 타입
- all-user-attribute-types-and-values : 엔트리에 연관된 모든 사용자 정보
- attribute-type : 특정 속성의 타입
- attribute-value : 특정 속성의 값
- all-attribute-values : 특정 속성에 연관된 모든 정보
- self-values : 사용자 요청이 속성에 있는 사용자 이름과 동일할 때만 액세스를 허용하는 속성값(사용자 이름과 일치하는 속성값일 때만 적용)

user class는 액세스 제어 명령이 적용되는 디렉토리 사용자들을 정의하는 것으로써 다음과 같은 항목들이 선택될 수 있다.

- all-users : 전제 디렉토리 환경의 모든 사용자
- this-entry : 접근되고 있는 엔트리와 동일한 고유 이름을 갖는 사용자
- name : 특정한 고유 이름을 갖는 사용자
- user group : 특정 고유 이름에 의해 식별되는 사용자 집합
- subtree : DIT의 서브트리 정의안에 속하는 사용자 집합

5.2 액세스 허가 범주

액세스 허가 범주는 적용할 액세스 제어 대

상에서 어떤 동작을 수행하기 위한 권한의 명시적 허가 및 거부에 관한 명세이다. 허가 범주는 엔트리 또는 속성 레벨에 적용할 것인지에 따라 구분할 수 있으며, 또한 어떤 허가는 특정의 디렉토리 요청 오퍼레이션에 대해서만 의미가 있을 수도 있다. 엔트리 레벨에서 사용될 수 있는 허가 범주에는 Read, Browse, Add, Remove, Modify, Rename, Disclose-OnError, Export, Import, Return_DN이 있다.

속성 레벨에서 사용될 수 있는 허가 범주에는 Compare, Read, FilterMatch, Add, Remove, DiscloseOnError가 있다.

5.3 액세스 제어 순위

액세스 제어 명령을 단계별로 적용하기 위하여 적용 우선 순위값, 인증 레벨, 그리고 다중 액세스 제어 명령에 대한 적용 절차 등의 정책이 포함되어야 한다.

적용 우선 순위값은 다수의 액세스 제어 명령이 적용될 때 우선 순위를 나타내는 0부터 255까지의 정수값이다. 명시된 요소가 모든 같은 경우에 높은 숫자가 낮은 숫자로 표현된 액세스 제어보다 상위 우선 순위를 갖는다.

인증 레벨은 액세스제어 명령을 적용하기 위한 결정 과정에서 인증 절차의 강도를 나타낸다. 기본 인증레벨은 non, simple, strong의 3가지 표준이 정의되어 있으며 이외의 다른 레벨이 표준 밖에서 정의될 수 있는 단계가 있을 수 있다. none은 어떤 사용 가능한 정보를 공개적으로 읽을 수 있는 경우로써 인증이 필요없음을 나타내고, simple은 개인의 사용자 속성에 대한 허가를 결정할 때 적용될 수 있다. 그리고, strong 인증 같은 경우는 보안 관리 정보를 수정하는 허가를 정의할 때 필수적으로 고려해야 할 것이다.

액세스 제어의 적용 절차는 액세스 제어 결정 함수(ACDF : Access Control Decision Function)라고 부르는 기본 알고리즘에 의하여 제어 결정이 이루어지는 방법에 대한 명세이다. 즉, 주어진 보호 항목에 대하여 특정 요청자에게 명시된 허가를 허용하거나 또는 거절할 것인지를 결정하기 위하여 ACI 항목들이 처리되는 방법을 기술한다. ACDF의 각 호출은 요청자의 고유이름, 보호 대상(엔트리, 속성 타입 또는 값), 액세스 허가 범주, 적용 우선 순위 정수 또는 인증 레벨 등의 ACI 보호 항목들로 이루어진다.

5.4 액세스 제어 범위

DIT의 특정 영역을 공통의 액세스 제어문이 적용되는 별도의 제한된 범위로 구분하는 것이 필요한데 이러한 영역을 관리영역(Administrative Area)라고 한다. 관리영역에는 ACSA(Access Control Specific Area)와 ACIA(Access Control Inner Area) 2 가지 유형이 있다[5][6][9].

디렉토리 관리 모델은 하나의 기관에 의하여 자율적으로 관리되는 AAA(Autonomous Administrative Area)으로 DIT를 구분한다. 또한, AAA는 ACSA라고 하는 서브영역으로 겹치지 않게 구분될 수 있으며 (그림 10)은 AAA를 3개의 ACSA로 구분한 예를 나타내고 있다. 이 예에서 A, B, C는 각각 ACSA의 서브트리 루트로써 그 영역의 관리 엔트리(Administrative Entry)라고 부른다. 이 관리 엔트리에 연관되어 있는 것이 서브 엔트리로써 그 영역내에 규정된 액세스 제어 정보를 이 곳에 갖고 있다. (그림 9)에서 영역 2에 있는 엔트리는 엔트리 B의 서브 엔트리에 규정된 액세스 제어 정보에 적용을 받는다.

ACIA는 ACSA가 더욱 구분된 것으로써 관리 엔트리와 서브 엔트리를 갖고 있는 점은 ACSA와 동일하다. 그러나, ACIA의 엔트리는 자신의 서브 엔트리에 정의된 액세스 제어에

종속됨과 동시에 또한 확장하여 ACSA로 둘러쌓인 액세스 제어에도 적용을 받는다. 즉, 영역 5의 엔트리는 A, D, E의 서브 엔트리에 규정된 액세스 제어 정보에 적용을 받는다.

ACSA와 ACIA는 관리 엔트리의 서브 엔트리에 규정된 액세스 제어 정보 명령문의 적용 범위에 관한 사항이다. 그러나, 명령이 명령내의 모든 엔트리에 적용될 필요는 없다. 즉, 모든 서브 엔트리는 DACD(Directory Access Control Domain)을 식별하는 서브트리 명세를 갖도록 하여 대상 엔트리를 정확히 어떤 액세스 제어 명령에 적용할지 식별하게 할 수 있다. 관리 영역내에는 몇 개의 DACD가 다중으로 중복될 수 있으며 (그림 10)은 2개의 DACD가 겹친 예를 나타내고 있다. 이 예에서 E1은 두 개의 영역에 모두 종속되므로 양쪽 관리영역의 각 서브 엔트리에 규정된 명령에 적용을 받는다.

6. 결 론

X.500 디렉토리 표준은 전세계적인 규모의 정보통신망을 상호 연결하는데 있어서 다목적 분산 디렉토리 서비스를 구축하기 위한 기반 기술을 제공한다. 디렉토리는 수많은 사람들 또는 통신 네트워크 및 기타 시스템에게 하나

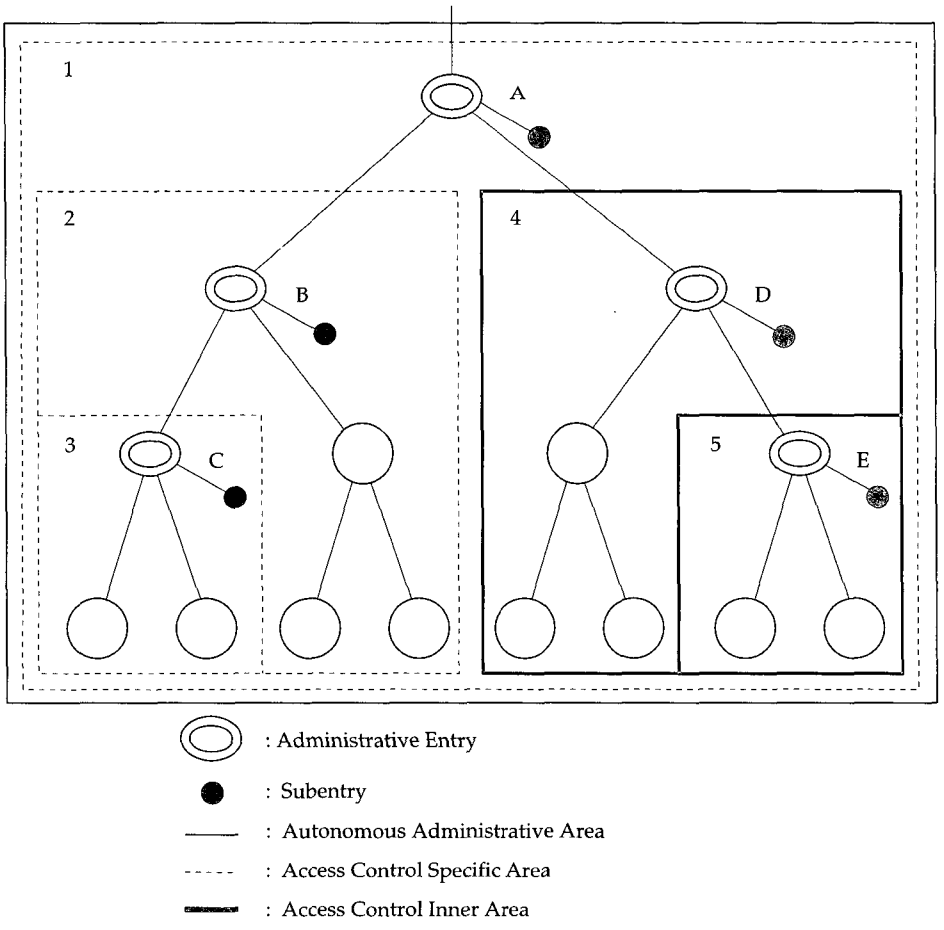


그림 9 액세스 제어 영역의 분할

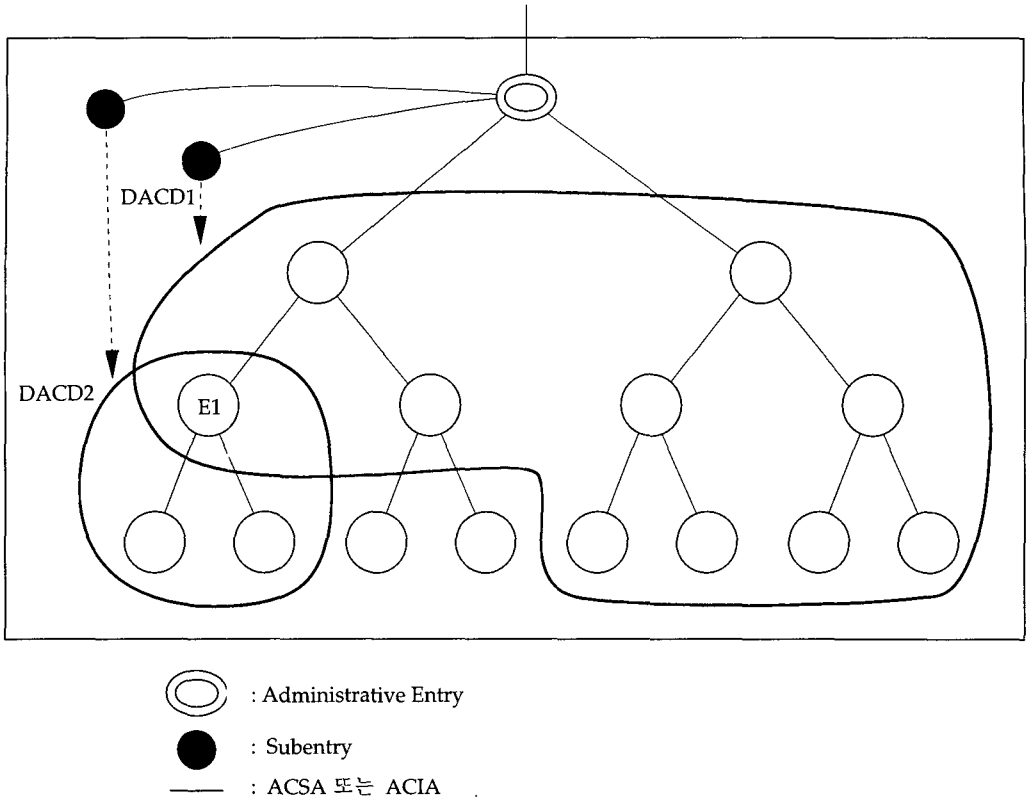


그림 10 디렉토리 액세스 제어 영역

의 정보 근원지로써 작용하며, 여러 형태의 기관 또는 개인에 대하여 다양한 형태의 정보 서비스가 가능하다.

그러나, EDI(Electronic Data Interchange)와 같은 고도의 신뢰성을 요구하는 응용 분야에서는 기밀성, 무결성, 실체인증, 부인봉쇄 등의 자체 정보 보호 정책이 전체적 디렉토리 서비스 표준 모델과 일치하도록 시스템이 구축되어야 한다. X.435 권고에서는 EDI응용에 디렉토리의 역할을 위하여 표준화 골격이 고려되어 있다. 즉, EDI User, DEI User Agent, EDI Message Store 와 같은 새로운 객체 클래스의 정의를 이용하여 naming 문제를 효율적으로 해결하거나 StrongAuthenticationUser 및 Certification-Authority와 같은 기존의 객체 클래스를 활용하여 secure EDI을 제공하는데 공개키 암호시스템의 사용을 용이하게 할 수 있다.

또한, X.509 인증 골격에 제시된 기법들을 사용하여 디렉토리가 인증서 형태로 공개키 정보의 보관소로써 사용될 수 있다. 이것은 EDI 자료에 전자서명을 부가하는 것을 포함하여 메시지 배달을 위해 사용된 메시지 처리 시스템의 전체적 정보보호 기능을 EDI응용 프로그램이 사용하도록 할 수 있다. 즉, 디렉토리 표준이 정보보호에 관련된 자료들의 분배 메카니즘으로 사용될 수 있는데, 이것은 EDI 자료에 서명을 하고, 인증 서비스를 제공하는데 이용될 수 있을 것이다. 따라서, EDI 관련 정보를 저장하기 위해 전세계적인 X.500 디렉토리의 사용이 가능하도록 설계하고 이것을 EDI처리에 밀접하게 통합하는 것은 여러가지 잇점을 얻을 수 있게 할 것으로 기대된다. 그리고, 앞으로 완전하고 안전한 EDI 서비스를 위해서는 EDIFACT(EDI for Administnation,

Commerce, and Transport), X.400 메시지 처리 시스템 및 X.500 디렉토리 서비스 등 수많은 네트워킹 기술들 모두가 연합될 수 있을 때 비로서 폭 넓은 응용 분야에서 활용이 보장될 것으로 전망된다.

그러나, 여러가지 경우를 고려한 방대하고 복잡한 모든 표준안 자료들을 정확히 분석하여 반영한다는 것은 실제로 어려운 일이다. 더구나, 기술적 수준이 낮아서 표준안 자체의 어떤 결합을 발견 못하거나 또는 적용 범위 및 기술이 불완전하다면 큰 문제일 것이다. 따라서, 응용분야를 근거로 한 자체 정보 보호 정책을 완전하게 수립하고, 실제적 표준안 적용 범위를 솔직하고 명확하게 밝힐 필요가 있으며, 정보 보호 서비스 기술 연구개발에 대한 지속적 관심과 투자가 요망된다.

참 고 문 헌

[1] 한국정보과학회, "정보과학회지 : 특집 - 초고속 정보통신용 소프트웨어, 보안 S/W", 제13권 제2호, 1995. 2.

[2] William Stalling, Network and internetwork Security, Prentice - Hall, 1995.

[3] Uyless Black, Network Management Standards, McGraw - Hill, 1995.

[4] George Coulouris, Jean Dollimore, Tim Kindberg, Distributed Systems Concepts and Design, 2nd ED, Addison Wesley, 1994.

[5] Warwich Ford, "Computer Communications Security"(Principles, Standard Protocols and Techniques), PTR Prentice Hall, 1994.

[6] CCITT Recommendation X.500, X.501, X.509, X.511, X.518, X.519, X.520, X.521, 1988, 1992.

[7] P.J. Bumbulis, D.D. Cowan, C.M. Durance, T.M. Stepien, "An Introduction to the OSI Directory Services", Computer Networks and ISDN Systems, 26 (1993), pp. 239-249.

[8] Bohdan Smetaniuk, "Distributed operation of the X.500 directory", Computer Networks and ISDN Systems 21 (1991), pp. 17-40.

[9] 최용락, "X.500 디렉토리 정보보호 서비스", 안전한 EDI 관련 기술 심포지움, 1995. 7. 6, pp. 121-143.

[10] 이재광, 이용준, "X.500 디렉토리 정보보호", 통신정보보호학회지, 제4권 제3호, 1994. 9, pp. 22-33.

[11] CCITT Recommendation X.435, Message Handling Systems : EDI Messaging System, Geneva, 1991.

[12] S.Kille, Implementing X.400 and X.500 : The PP and QUIPU Systems, Artech House, Inc. ISBN 0-8900, 6-564-0, 1991.

[13] Hossam Afifi, "A hierarchical directory based X.400 server", Compuer Networks and ISDN Systems 26 (1993), pp. 317-326.

[14] C. I'Anson, and C. Mitchell, "Security Defects in CCITT Rec. X.509 - The Directory Authentication Framework", Computer Communications Review, April 1990.

[15] Donal O'Mahony and Neil Weldon, "X.500 directory services support for

Electronic Data Interchange", Computer Networks and ISDN Systems 27(1995), pp. 691-701

[16] Judith King, "X.400 Security", Computers & Security, 11(1992), pp. 707-710.

[17] ATH van der Voort, A L Boonstra and FHR de Thouars, "Routing names and addresses in the international MHS, Computer Communications Vol 18, no.4, Apr. 1995, pp. 247-251.

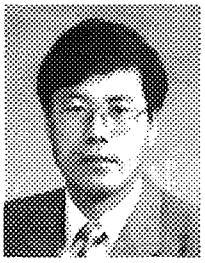
[18] J. Carr, "EDI - Security Risk or Not?", Computers & Security, 10(1991), pp. 69-72.

□ 著者紹介

최 용 락

1976년 2월 중앙대학교 전자계산학과(학사)
1982년 2월 중앙대학교 대학원 전자계산학과(석사)
1989년 2월 중앙대학교 대학원 전자계산학과(박사)
1982년 ~ 1986년 한국전자통신연구소 선임연구원
1986년 ~ 현재 대전대학교 컴퓨터공학과 교수

※ 주관심분야 : 운영체제, 컴퓨터/네트워크 보안



강 창 구

1979년 한국항공대학 항공전자공학과 졸업 (공학사)
1986년 충남대학교 대학원 전자공학과 (공학석사)
1993년 충남대학교 대학원 전자공학과 (공학박사)
1979년 ~ 1982년 한국공군 기술장교
1987년 ~ 현재 한국전자통신연구소 책임 연구원

김 대 호

1977년 한양대학교 전자공학과(학사)
1984년 한양대학교 산업대학원 전자공학과(석사)
1993년 Univ. of Maryland at College Park
Dept. of Computer Science Visiting Scholar
1977년 ~ 현재 한국전자통신연구소 책임연구원

※ 주관심분야 : 전송분야, 통신 및 컴퓨터 보안