윈도우 운영 체제에서 불법 소프트웨어 방지 시스템

황 기 태† · 김 남 윤††

요 약

본 논문은 조직 내에 소프트웨어의 불법적인 설치/사용을 방지하는 소프트웨어 시스템에 관해 기술한다. 이 시스템에서 서버 컴퓨터는 모든 소프트웨어를 압축하여 관리하며, 클라이언트 컴퓨터는 반드시 서버로부터 소프트웨어를 다운로드 받아 설치하고 불법 소프트웨어의 설치 상태를 주기적으로 서버에 보고한다. 또한 외부로의 정품 소프트웨어의 유출과 변조를 방지하기 위해 윈도우 운영체제에서 인증 시스템과 세션 키를 이용한 암/복호화 개념을 도입한다. 본 논문에서 제안된 시스템은 정품 소프트웨어의 불법 사용 방지뿐만 아니라 라이센스 통제와 소프트웨어 관리의 용이성의 장점을 가진다.

System for Anti-Piracy of Software under Windows Operating System

Kitae Hwang† · Namyun Kim††

ABSTRACT

This paper presents the software system that protects illegal installation and use of the commercial software. The server computer in this system stores the compressed versions for all software, while client computers install all software by downloading them from the server. Also the client computers periodically report to the server whether they have illegally installed software. This system introduces authentication and encryption/decryption using the session key under Windows Operating System to prevent interception of the software package from outside world and malicious modification of the transfer message between the server and the client. The proposed system in this paper has several advantages such as providing real-time control of license and easy maintenance of the software as well as protecting illegal use of the software.

키워드 : 불법 소프트웨어(Anti-Piracy of Software), 보안(Security), 레지스트리(Registry), 인증(Authentication), 암호화(Encryption)

1. 서 론

우리나라 정보화 산업은 매우 빠르게 발전해 가고 있다. PC 보급대수 2500만대, 1000만 고속 인터넷 가입자라는 세계 최고의 기반(infrastructure)이 불과 몇 년 사이에 이루어져, 산업적인 측면에서 뿐만 아니라 개인적인 면에서도 많은 긍정적인 변화를 유발하였다. 그러나 급속한 발전에 따른 부작용이라고 할 수 있는 문제들도 동시에 발생했다. 그 가운데 주목할만한 부작용은 소프트웨어 불법 복제에 따른 지적 재산권 침해와 관련된 것이다. 2002년 12월 기준으로 불법 소프트웨어로 인한 피해 금액은 약 139억원에 이르며, 이러한 손실은 결국 소프트웨어 개발을 저해하는 부메랑으로 돌아오게 되는 문제점을 낳게 된다[1].

현재 불법 소프트웨어의 사용을 막기 위한 많은 방법들이 개발되어 있다. 이러한 방법은 불법 복제를 근본적으로 막는 방법과 불법 소프트웨어의 설치나 사용을 차단하는 방법으로 구분된다. 첫째, 불법 복제를 막는 방법으로서 CD 복사 방지 방식은 광 디스크의 물리적인 특성과 CD-ROM/DVD-ROM Drive의 디스크 구동 메커니즘을 응용한 OHD(Optical Head Ditching) 기술을 적용하여 처음부터 복사가 불가능한 CD에 소프트웨어를 제작하는 방법이다[2]. 또 다른 방법인 CD-COPS라는 기술 역시 CD 표면에 미세한 차이의 핑커 프린트(finger print)를 해놓고 특별한 접근 코드(access code)를 가진 핑커 프린트를 읽어 들이게 함으로써 어떤 수단으로도 불법 복제가 되지 않게 하는 방법이다[3]. 다른 방법으로는 USB 포트나 병렬 포트에 메모리나 프로세서를 장착한 하드웨어 키 모듈을 설치하고키 모듈에 키 데이터나 암호화 알고리즘을 구현하여 하드웨어 키가 없이 소프트웨어를 컴퓨터에 설치할 수 없도록 하는 방법이 있다. 둘째, 소프트웨어의 설치를 관리함으로써 불법 소프트웨어의 사용을 막는 방법으로 주로 소프트웨어적으로 이루어진다. 이 방법은 클라이언트 컴퓨터에 특정 소프트웨어를 구동하여 설치된 소프트웨어 리스트를 관리용 컴퓨터

※ 본 연구는 2004학년도 한성대학교 교내 연구비 지원 과제임.
† 정 회 원 : 한성대학교 컴퓨터공학부 교수
†† 정 회 원 : 한성대학교 정보공학부 교수
논문접수 : 2003년 9월 2일, 심사완료 : 2003년 12월 10일

에게 주기적으로 전송한다[4, 5].

불법 소프트웨어의 사용을 막는 이러한 방법들은 몇 가지 문제점을 가지고 있다. 하드웨어 키를 사용하여 불법 복제를 막는 방법은 소프트웨어의 판매 단가가 올라가기 때문에 소프트웨어 제작사들이 이 방법들을 거의 채택하지 않고 있다. 클라이언트 상에 설치된 불법 소프트웨어를 서버가 단순 모니터링하여 불법 소프트웨어의 설치를 가려내는 방법 또한 CD의 불법 복사 및 유출, 분실의 가능성, 재설치시의 번거로움, 라이센스(license)의 카피 수에 대한 실시간 통제가 되지 않는 점 등의 문제점을 해결하지 못한다.

본 논문은 불법 소프트웨어의 설치를 감시하는 새로운 방법을 제안하고 설계 구현한 내용을 다루고자 한다. 본 논문에서 설계한 시스템은 소프트웨어의 불법 복제를 막기 위한 것이 아니라, 한 조직 내에 불법 소프트웨어가 설치되어 사용되는 것을 막기 위한 것으로서 두 가지 접근 방식을 가진다. 첫째, 클라이언트 컴퓨터는 대리(agent) 프로그램을 이용하여 서버로부터 정식 소프트웨어를 다운 받아 설치한다. 이 때 소프트웨어의 불법 유출을 막기 위해 인증, 암호화와 같은 보안 기법을 사용한다. 따라서 그룹 내에서 정품 소프트웨어의 복사본을 만들 필요가 없으므로 정품 소프트웨어의 유출 가능성이나 정품 소프트웨어의 원본 CD의 분실 가능성이 없으며, 라이센스 카피 수의 실시간 통제가 가능하다는 장점을 가진다.

둘째, 클라이언트 컴퓨터상의 대리 프로그램은 불법으로 설치된 소프트웨어를 모니터링하여 서버로 통보한다. 따라서 서버는 조직 내의 클라이언트 컴퓨터에 설치된 모든 소프트웨어를 파악함으로써 불법 소프트웨어 설치 여부를 파악할 수 있고 소프트웨어 통계 자료도 쉽게 유도할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 연구의 배경 지식을 논하며 3장에서는 시스템의 구성 및 보안 설계를, 4장에서는 소프트웨어 설계에 관한 내용을 논한다. 5장에서는 구현 결과를 보이고 6장에서 결론을 맺는다.

2. 연구 배경

본 절에서는 클라이언트가 서버 컴퓨터로부터 소프트웨어를 전송 받을때 외부로의 유출 및 변조 방지를 위해 도입되는 보안 기술을 설명한다. 그리고 클라이언트 컴퓨터의 모니터링을 통한 불법 소프트웨어 감시 기능을 위해 필요한 윈도우 레지스트리에 대해 기술한다.

2.1 소프트웨어 보안

2.1.1 보안 서비스

본 논문에서 클라이언트와 서버 컴퓨터간에 소프트웨어 전송 시 유출 및 변조 방지를 위해 제공되는 보안 서비스는 다음과 같다[6].

① 기밀성(Confidentiality) : 정당한 권한을 가진 클라이언트만이 소프트웨어를 설치할 수 있도록 하기 위해 소프트웨어를 암호화한다. 따라서 외부에서 전송되는 메시지를 도청하더라도 해독할 수 없기 때문에 소프트웨어가 불법적으로 사용되는 것을 막을 수 있다.

② 인증(Authentication) : 클라이언트의 신원을 확인하는 서비스이다. 서버로부터 인증 받은 클라이언트만이 서버로부터 소프트웨어 전송 후 설치할 수 있도록 함으로써 소프트웨어의 유출을 막을 수 있다.

③ 무결성(Integrity) : 전송 과정에서 소프트웨어의 변조 유무를 확인하는 서비스이다. 외부로부터 소프트웨어 변조 시 클라이언트 컴퓨터에 설치가 불가능하기 때문에 변조 유무를 미리 파악함으로써 안전한 소프트웨어 설치를 보장할 수 있다.

2.1.2 세션 키를 이용한 암호화

기밀성 제공을 위한 암호화 알고리즘에는 대칭 키 및 공개키 암호 알고리즘이 존재한다[7]. 대칭 키 암호 알고리즘은 공개키 암호 알고리즘보다 암/복호화 속도가 빠르기 때문에큰 메시지의 암호화 시에 사용된다. 한편 대칭 키 암호는 송/수신자간에 공유되는 하나의 대칭 키(symmetric key)를 이용하여 암/복호화하며, 대칭 키를 매 세션마다 다르게 설정함으로써 안전성을 높인다. 이와 같이 세션마다 사용되는 대칭 키를 세션 키(session key)라고 한다. 본 논문에서는 전송되는 소프트웨어 암호 시 세션 키를 이용한 대칭 키 암호 알고리즘을 이용한다.

2.1.3 세션 키 분배와 인증

대칭 키 암호 알고리즘은 송/수신자간에 미리 세션 키가 공유되어 있어야 한다. 따라서 세션 키를 미리 안전하게 분배하는 "키 분배 문제"를 해결하여야 한다. 키 분배 문제를 해결하기 위한 한 방법으로서 공개 키 암호 알고리즘이 자주 사용된다. 공개 키 암호 알고리즘에는 한 쌍의 공개 키(public key)와 개인 키(private key)가 사용되는데, 공개 키를 이용하여 암호화된 메시지가 개인 키를 이용하여 복호화되며 역으로 개인 키를 이용하여 암호화된 메시지는 공개 키를 이용하여 복호화될 수 있다. 사용자는 공개 키와 개인 키 쌍을 생성한 후 개인 키는 안전하게 보관하고 공개 키는 외부 세계에 공개한다. 일반적으로 세션 키를 분배하는 방법은 다음과 같다. 서버가 세션 키를 생성한 후 클라이언트의 공개 키를 가지고 세션 키를 암호화한다. 그리고 암호화된 세션 키를 클라이언트에게 전달한다. 클라이언트는 자신이 보관하고 있는 개인 키를 이용하여 암호화된 세션 키를 복호화하여 세션 키를 얻는다.

사용자의 신원을 확인하는 인증 방법으로는 "사용자 이

름/비밀 번호", "시도/응답 프로토콜" 등이 존재한다[6]. 시도/응답 프로토콜은 클라이언트가 자신의 신분을 서버에게 증명해 보이기 위해, 자신만이 소유하고 있는 비밀 정보를 알고 있다는 사실을 상대방에게 간접적으로 보여주는 프로토콜이다. 예를 들면, 클라이언트가 서버측에서 보내온 난수를 자신의 개인 키로 암호화한 디지털 서명한 결과를 보내고 서버는 클라이언트의 공개 키를 이용하여 서명을 확인하여 클라이언트가 개인 키를 알고 있음을 간접적으로 증명할 수 있다.

한편, 사용자의 공개 키는 공개되어 있지만 변조되었을 경우 쉽게 판별할 수 있어야 한다. 이를 위하여 사용자의 공개키는 인증서의 형태로 보관된다[8]. 인증서는 인증 기관에 의해 발급되며, 공개 키와 사용자의 정보를 바탕으로 인증 기관이 디지털 서명을 생성하여 인증서에 저장한다. 디지털 서명 값을 확인해 봄으로써 공개키 변조 유무를 판별할 수 있다.

2.1.4 해쉬 함수

메시지의 무결성을 확인하기 위해 해쉬 함수가 사용된다. 해쉬 함수는 메시지의 길이에 상관없이 항상 고정 길이의 해쉬 값을 생성하며, 대표적인 해쉬 함수로는 MD5[9], SHA-1[10]이 존재한다. 송신자는 메시지와 그 해쉬 값을 함께 전송하고, 수신자는 메시지를 바탕으로 해쉬 값을 계산하고 수신된 해쉬 값과 비교함으로써 메시지의 변조 유무를 판단할 수 있다.

2.2 윈도우 레지스트리

윈도우 레지스트리(Registry)란 Windows 98/98/2000/XP 등에서 운영체제가 실행되는데 필요한 모든 정보를 가진 일종의 데이터베이스이다. 레지스트리에는 컴퓨터에 설치된 하드웨어, 소프트웨어, 사용자, 네트워크 설정, 드라이버, 환경 변수 등의 정보들이 포함되며 이들 정보를 6개의 그룹으로 나누어 관리된다. 각 그룹은 트리(tree) 구조로 형성된 키(key)들로 조직화되어 있다. 현재 컴퓨터 상에 설치된 모든 소프트웨어에 관한 정보들은 HKEY_LOCAL_MACHINE이라는 키를 루트 키 혹은 주 키(main key)로 하는 정보 그룹에 저장되어 있으며 구체적으로 주 키 밑의 SOFTWARE라는 부 키(sub key)에 저장되어 있다. 이 키 값을 읽어 현재 설치된 소프트웨어 목록을 얻어낼 수 있으며 이를 통해 불법으로 설치된 소프트웨어의 설치 여부를 판단할 수 있다.

3. SAP 시스템 모델

3.1 시스템 구조

본 논문에서 제안된 시스템은 SAP(System for Anti-Piracy)이라고 명명되며 그 구조는 (그림 1)과 같이 구성된다. SAP은 기본적으로 다수의 컴퓨터와 고속 네트워크를 가진 조직(Organization)에서 조직내의 구성원들이 사용하는 컴퓨터에 불법 소프트웨어가 설치되지 않도록 감시하며 정품 소프트웨어를 자동으로 설치할 수 있도록 관리하는 시스템 모델이다. SAP 시스템은 다음과 같이 7가지의 구성 요소로 정의된다.

- Server - 서버 컴퓨터
- Client - 조직 내의 클라이언트 컴퓨터
- PCA(Private Certificate Authority) - 사설 인증 서버
- User - 조직의 구성원으로서 클라이언트 컴퓨터의 사용자
- Manager - Server 및 시스템 관리자
- IMS(Installation Management Server) - Server에서 실행되는 소프트웨어 모듈
- MAP(Monitoring Agent Program) - Client에서 실행되는 소프트웨어 모듈

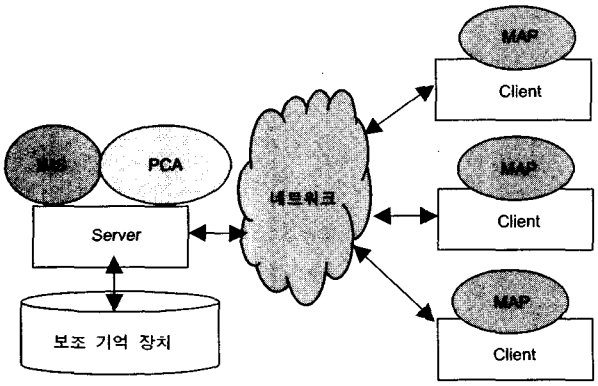
SAP을 구성하는 하드웨어 요소는 Server와 Client이며 인적 요소로는 User와 Manager, 소프트웨어 요소로 PCA, IMS, MAP 등이다. IMS와 MAP은 Server와 Client에서 항상 실행중인 상태로 설정되며 PCA는 Client에 인증서를 발급하고 인증서를 관리하는 소프트웨어 요소로서 Server에서 실행된다. Server는 대용량의 하드 디스크를 가지며 조직 내의 모든 정품 소프트웨어를 압축하여 저장하고 시스템을 관리하기 위한 DB 테이블을 관리 유지한다.

SAP 시스템에서 소프트웨어 설치 과정은 다음과 같다.

① Client의 인증서 설치 : User 혹은 Manager가 Client를 초기 세팅할 때 Server의 PCA에 접속하여 인증서를 발급 받아 Client 컴퓨터 상에 저장한다.
② Client 인증 : User가 Client에 소프트웨어를 설치하고자 하는 경우 MAP을 이용하여 Server의 IMS에 접속하고 인증 절차를 거친다.
③ 세션 키 분배 : Server의 IMS는 세션 키를 생성하고 Client의 인증서에 포함된 공개 키로 세션 키를 암호화하여 Client의 MAP에게 전송한다. MAP은 이 메시지를 개인 키로 복호화하여 세션 키를 확보한다.
④ 소프트웨어 요청 : Client의 MAP은 정품 소프트웨어 리스트를 요청하여 전송 받은 후 설치를 원하는 소프트웨어를 요청한다.
⑤ 암호화된 소프트웨어 전송 : Server의 IMS는 세션 키로 소프트웨어를 암호화하여 MAP에게 보낸다.
⑥ 소프트웨어 설치 : Client의 MAP은 세션 키로 소프트웨어를 복호화하고 Client에 설치한다. 설치가 종료되

는 즉시 MAP은 세션 키와 전송 받은 데이터를 모두 삭제하여 정품 소프트웨어의 유출을 막는다.

그리고 Server의 IMS는 주기적으로 Client의 MAP로부터 설치된 소프트웨어 리스트를 전송받아 불법 소프트웨어를 감시한다.



(그림 1) SAP 시스템 모델

3.2 보안 설계

SAP 시스템은 시스템의 보안을 유지하기 위해 두 가지 보안 정책을 가지고 있다. 하나는 유출 보안으로서 Server가 가진 정품 소프트웨어가 조직 내의 사용자나 외부의 공격자에게 유출되지 않도록 하는 정책이며, 두 번째는 변조 보안으로서 Client 상의 MAP과 IMS의 통신 동안 내부나 외부의 악의적인 사람들로부터 데이터 변조를 막는 보안 정책이다. 유출 보안을 위해서는 조직의 구성원임을 확인하는 인증 메커니즘과 암호화 기법을 사용한다. 그리고 변조 보안을 위해서는 통신 데이터에 해쉬 값을 추가하여 무결성을 확인한다.

3.2.1 세션 기반 보안

IMS와 MAP은 세션(session)을 기반으로 동작한다. 세션이란 MAP과 IMS가 보안을 유지하며 통신하는 작업 단위로 정의된다. SAP에는 두 가지 세션이 존재한다. 첫째는 설치 세션(installation session)으로서, 소프트웨어를 설치하고자 할 때 MAP에 의해 시작되는 세션이며 설치가 종료될 때까지 지속된다. 둘째는 보고 세션(reporting session)으로서 MAP이 불법 소프트웨어 설치 여부를 IMS에 보고하는 세션이다.

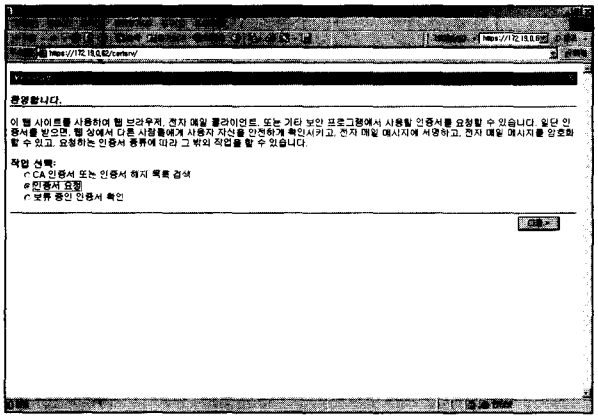
3.2.2 인증서 발급

조직의 구성원이 아닌 사람이 SAP 시스템의 정품 소프트웨어나 기타 정보들을 유출하는 것을 방지하기 위해서는 세션의 시작 시점에서 MAP에 대한 인증이 필요하다. 본 논문에서는 공개키 암호 알고리즘에 기반한 "시도/응답 프로토콜"을 이용하여 인증을 수행하기 때문에 각 사용자는 인증서를 발급받고 개인 키를 안전하게 보관하여야 한다.

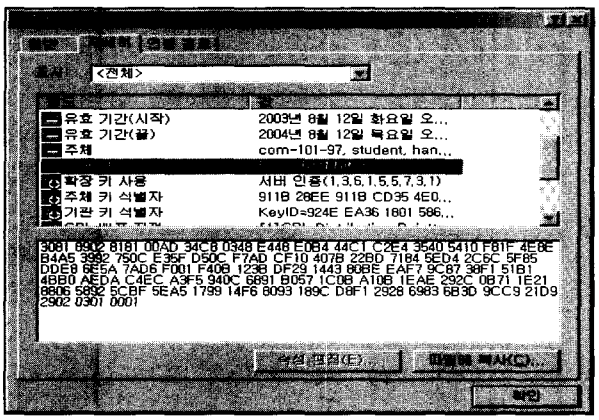
SAP 모델에서는 사설 인증 기관(PCA)을 Windows 2000 Server에 설치하였다. User나 SAP 시스템 Manager는 Client에 MAP을 설치하기 전에 먼저 인증 서버(PCA)로부터 사설 인증서를 발급 받는다.

인증서가 발급되는 과정은 다음과 같다. Manager는 웹으로 Server에 접속하여 인증서를 요청한다((그림 2)(a)). 그리고 나서 Manager는 client ID와 필요한 정보를 입력하고 인증서 발급을 신청한다. Server가 클라이언트가 요청한 인증서를 승인하면 Manager는 서버로부터 인증서를 수신한 후 Client 컴퓨터에 설치한다.

발급된 인증서는 X.509 포맷으로서 공개키, client ID 정보, 그리고 PCA의 서명 값을 가지고 있다((그림 2)(b)). PCA는 자신이 발급한 인증서를 디스크 상에 저장하여 관리하며 IMS는 인증을 위해 필요할 때 인증서를 액세스할 수 있다.



(a)



(b)

(그림 2) 인증서 요청 및 인증서 형식

3.2.3 인증(Authentication)

Server의 IMS가 MAP으로부터 세션 연결 요청시, 정당

한 사용자인지를 확인하는 과정이다. SAP 시스템은 "시도/응답 프로토콜"을 이용하여 Client를 인증하며 인증 과정은 다음과 같다.

① Client의 MAP은 client ID를 IMS에 보낸다. client ID는 클라이언트 컴퓨터가 인증서를 발급 받을 때 입력한 정보이다.
② Server의 IMS는 인증 서버가 관리하는 인증서 저장소에서 client ID 값을 가진 인증서를 찾는다.
③ Server의 IMS는 난수(random number)를 생성하고 MAP으로 보낸다.
④ Client의 MAP은 수신한 난수 값을 자신의 개인 키로 암호화하여 IMS로 보낸다.
⑤ Server의 IMS는 수신한 암호화 메시지를 인증서에 저장된 공개 키를 이용하여 복호화하고 자신이 보낸 난수와 일치하는지 비교한다. 만일 동일하면 MAP이 개인 키를 알고 있음이 증명되므로 인증 과정이 종료된다.

3.2.4 세션 키 분배

SAP 시스템에서는 세션의 보안을 위해서 세션 키를 사용한다. 세션이 시작될 때 Server의 IMS는 세션 키를 생성하고 세션이 종료되면 소멸되므로 외부로 세션 키가 유출되더라도 다음 세션에서 그 세션 키는 의미가 없게 된다. SAP 시스템에서 세션 키를 생성하고 분배하는 과정은 다음과 같다.

① Server의 IMS는 세션 연결 요청시 MAP 인증을 수행한다.
② Server의 IMS는 세션 키를 생성한 후, MAP의 공개 키로 암호화하여 전송한다.
③ Client의 MAP은 암호화된 메시지를 자신의 개인 키로 복호화하여 세션 키를 얻는다.

본 논문에서 사용된 대칭키 암호 알고리즘은 RC4[7]이며 세션 키는 128비트이다.

3.2.5 소프트웨어 암호화

일반적으로 소프트웨어의 용량은 매우 크기 때문에 세션 키로 암호화하는데 상당한 시간이 소요된다. 본 논문에서는 송수신 시 암/복호화 시간을 줄이기 위해 다음과 같은 방법을 도입한다. 먼저 소프트웨어를 ZIP[11]으로 압축한 후, 압축 파일에서 헤더 부분을 더미 데이터로 채워 압축 파일을 변경한다. 그리고 변경된 압축 파일과 원본 헤더를 디스크에 저장한다. 실제 전송 시에는 압축 파일의 원본 헤더와 기타 정보로 구성된 "전송 메시지"를 생성하여 세션 키로 암호화한 후, 압축 파일과 함께 Client의 MAP으로 전송한다. 따라서 "전송 메시지"만 암호화를 수행함으로써 시간을 단축시킬 수 있으며 압축 파일은 헤더 부분이 더미 데이터로 채워져 있으므로 도청이 되더라도 안전하다고 볼 수 있다. 한편, Client의 MAP은 "전송 메시지"를 세션 키로 복호화하여 원본 헤더를 얻는다. 그리고 원본 헤더를 바탕으로 압축 파일을 복구하여 소프트웨어를 설치한다. "전송 메시지"에 대한 자세한 내용은 4.1.4절에서 설명한다.

3.2.6 무결성

Server는 소프트웨어 전송시 변조 유무를 파악하기 위해 해쉬 함수를 사용한다. 3.2.5절에서 서술한 "전송 메시지"의 해쉬 값을 계산하여 전송 메시지에 저장한다. Client의 MAP은 전송 메시지를 세션 키로 복호화한 후 해쉬 값을 얻고, 전송 메시지로부터 직접 계산된 해쉬 값과 비교함으로써 무결성을 확인한다. 사용된 해쉬 알고리즘은 SHA-1[10]이다.

3.2.7 압축해제 후 소프트웨어 보안

코딩/압축된 소프트웨어가 Client에서 MAP에 의해 압축해제된 후 ZIP 파일이 외부로 유출될 가능성이 있다. 이 문제는 다음과 같이 해결된다. 첫째, 소프트웨어가 비록 유출된다고 하더라도 설치시 필요한 시리얼 키 값은 오직 MAP만이 가지고 있기 때문에 유출 후 설치가 거의 불가능하다. 둘째, Client 컴퓨터에서 MAP이 IMS로부터 소프트웨어를 받는 디렉토리는 MAP 만이 접근할 수 있도록 권한을 설정하면 다른 사용자나 프로세스가 ZIP 파일을 유출할 수 없다. 셋째, MAP은 설치가 끝나면 ZIP을 완전히 소멸시키기 때문에 유출의 가능성은 없다.

3.3 세션 설계

3.3.1 설치 세션(Installation Session)

Client 상의 User가 MAP 프로그램을 이용하여 소프트웨어를 다운로드 받고 설치하는 세션이다. 설치 세션 과정은 다음과 같다.

① Client의 MAP은 IMS로부터 인증을 받는다.
② Client의 MAP은 IMS로부터 세션 키를 분배 받는다.
③ Client의 MAP은 정품 소프트웨어 리스트를 IMS에게 요구하고 이 리스트를 받는다.
④ Client의 MAP은 Client 화면에 리스트를 출력하여 User에게 소프트웨어를 선택하게 하고 선택된 소프트웨어를 IMS에게 요청한다.
⑤ Server의 IMS는 압축 소프트웨어와 암호화된 전송 메시지를MAP에게 전송한다.
⑥ Client의 MAP은 세션 키를 이용하여 암호화된 전송 메시지를 복호화하여 무결성을 확인한다. 그리고 전송 메시지를 이용하여 압축 소프트웨어를 복원화한 후

특정 디렉토리에설치한다.

⑦ Client의 MAP은 설치가 종료되면 Client에 풀어져 있는 소프트웨어의 부분을 삭제하고 설치 결과를 IMS로 보낸다.

⑧ 세션을 닫는다.

3.3.2 보고 세션(Reporting Session)

Client의 MAP은 주기적으로 레지스트리를 검사하여 불법 소프트웨어가 설치되었는지 검사한다. 그리고 검사 결과를 Server의 IMS로 전송하게 되는데, 이를 보고 세션이라고 부르며 이 과정은 다음과 같다.

① Client의 MAP은 IMS로부터 인증 과정을 거쳐 인증을 받는다.

② Client의 MAP은 IMS로부터 세션 키를 분배 받는다.

③ Client의 MAP은 불법 소프트웨어 검사 결과를 세션 키로 암호화하여 IMS로 전송한다.

④ Server의 IMS는 암호화된 검사 결과를 세션 키로 복호화하고 불법 소프트웨어가 설치되었다면 이 결과를 관리자에게 보고하여 조치를 취하도록 한다.

⑤ Server의 IMS는 MAP에게 세션의 종료를 알린다.

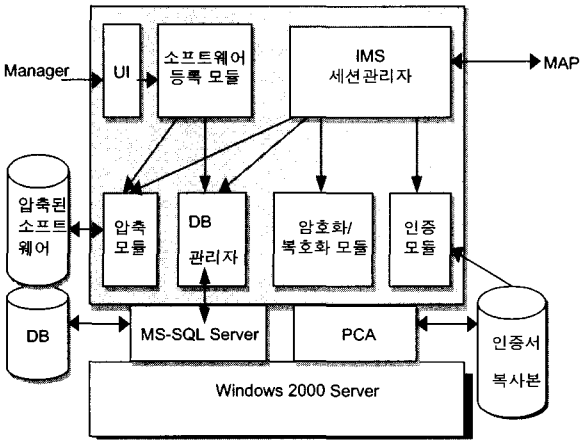
4. 소프트웨어 설계

4.1 IMS 설계

4.1.1 IMS 구조 설계

IMS 모듈의 내부 구조는 (그림 3)과 같이 IMS 세션 관리자, 소프트웨어 등록 모듈, 압축 모듈, DB 관리자, 암호화/복호화 모듈, 인증 모듈, 그리고 UI 모듈로 구성된다. IMS의 하부에는 MS-SQL Server가 설치되어 있으며, Windows 2000 Server의 한 부분인 인증 서버 PCA가 위치한다. IMS에 존재하는 데이터 저장소는 압축된 소프트웨어가 저장되는 저장소와 SAP 시스템의 관리를 위해 필요한 DB 테이블 저장소, 그리고 PCA에서 발급된 인증서의 복사본이 저장되는 저장소의 3부분으로 분류된다. 각 모듈에 대한 간략한 설명은 다음과 같다.

- IMS 세션 관리자는 MAP으로부터의 네트워크 접속과 함께 설치 세션과 보고 세션의 수행을 관리하는 기능을 수행한다.
- 소프트웨어 등록 모듈은 새로운 소프트웨어가 도입될 때 소프트웨어를 Server에 등록하는 기능을 수행한다.
- 압축 모듈은 소프트웨어 등록 모듈에 의해 호출되는 모듈로서 도입된 소프트웨어를 압축하는 기능을 제공한다.
- DB 관리자는 SAP 시스템에 필요한 DB 테이블에 대한 액세스를 관리하는 모듈이다.
- 암호화/복호화 모듈은 각 세션에서 MAP으로 전송할 메시지를 암호화하거나 MAP으로부터 전송된 메시지를 복호화하는 기능을 제공한다.
- 인증 모듈은 Client의 인증을 위해 PCA에 의해 발급된 인증서 복사본이 들어 있는 시스템 저장소에서 인증서를 찾고 읽는 기능을 제공한다.



(그림 3) IMS 내부 구조

4.1.2 IMS의 소프트웨어 분류

소프트웨어는 판매를 목적으로 하는 상업용 소프트웨어와 프리웨어(freeware)/쉐어웨어(shareware) 등과 같은 비상업용 소프트웨어로 구분된다. 비상업용 소프트웨어들은 불법 복제의 대상이 아니므로 개인 PC의 사용자가 임의로 설치하여 사용할 수 있다. 그러나 이들 비상업용 소프트웨어가 개인 PC에 설치되어 있을 때 불법이 아님에도 불구하고 MAP에 의해 불법 소프트웨어로 인식될 수 있다. 이 문제는 다음과 같이 해결한다.

Server의 IMS는 비상업용 소프트웨어에 대한 목록만을 유지하고 User는 자신이 원하는 비상업용 소프트웨어를 마음대로 설치 사용하는 방법이다. Client의 MAP이 비상업용 소프트웨어를 발견하고 불법 소프트웨어로 IMS에게 보고하게 되면 Server의 IMS는 MAP에게 그 소프트웨어가 비상업용임을 알려주는 방법이다. 만약 IMS가 관리하는 DB 테이블에 비상업용 소프트웨어가 등록되어 있지 않다면 Manager는 비상업용 소프트웨어로 등록하면 된다. 이 작업은 오프라인으로 이루어져도 전혀 문제가 없다.

4.1.3 IMS에 의해 관리되는 DB 테이블

IMS는 <표 1>과 같은 5개의 DB 테이블을 유지한다.

- client_list 테이블은 조직 내의 모든 Client에 관한 정보를 유지하는 테이블이다.
- sw_list 테이블은 IMS에 의해 관리되는 모든 소프트

웨어에 관한 정보를 담는 테이블이다. 비상업용 소프트웨어의 경우 오직 swName과 b 필드만이 유효하다.

- report_list 테이블은 MAP에 의해 불법 소프트웨어가 발견될 때마다 하나의 레코드가 생겨나는 테이블이다. 만일 한 Client에 3개의 불법 소프트웨어가 발견되었다면 총 3개의 레코드가 삽입된다.
- installed_list 테이블은 Client에 대해 설치된 모든 소프트웨어의 목록이 저장되는 테이블이다. 만일 어떤 Client에 10개의 소프트웨어가 설치되어 있다면 10개의 레코드가 존재할 것이다.
- installing_info 테이블은 현재 설치중인 소프트웨어에 관한 정보를 가지는 테이블로서, User가 다운로드 후 Client에 문제가 생기거나 고의적으로 설치를 지연하는 등의 문제점을 추적하기 위한 것이다.

〈표 1〉 IMS에 의해 관리되는 DB 테이블들

client_list 테이블

필 드	설 명
comID	Client의 고유 ID로서, 유일한 값. 현재는 컴퓨터 이름
tStart	Client에 MAP이 처음으로 설치되어 실행을 시작한 시간
comIP	Client의 IP 주소값
tReport	MAP의 의해 가장 최근에 불법 소프트웨어에 대한 보고가 이루어진 시간

sw_list 테이블

필 드	설 명
swName	소프트웨어 이름으로서 Manager가 입력한 이름
b	상업용 소프트웨어인지, 비상업용인지를 구분
originalSize	소프트웨어를 압축하기 전에 차지하는 용량
compressedSize	소프트웨어를 압축한 용량
time	소프트웨어를 준비한 시간
copies	소프트웨어가 설치된 Client의 개수
licenses	소프트웨어 설치 허용 개수
installFile	소프트웨어 설치 파일명
serialNum	소프트웨어의 설치 시리얼 번호
CDKey	소프트웨어의 설치 CD 키 값

report_list 테이블

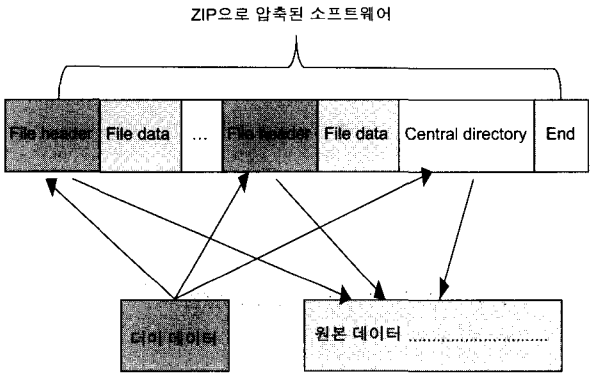
필 드	설 명
comID	Client의 이름
illegalSW	comID의 컴퓨터에서 발견된 불법 소프트웨어 이름
tFound	불법 소프트웨어를 발견한 날짜

installed_list 테이블

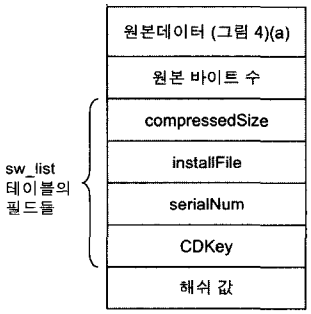
필 드	설 명
comID	Client의 이름
installedSW	설치된 소프트웨어의 이름으로서, 시스템 레지스트리에 기록된 이름
tInstalled	소프트웨어가 설치된 시간

installing_info 테이블

필 드	설 명
comID	Client의 이름
installingSW	현재 설치중인 소프트웨어의 이름
tInstall	설치를 시작한 시간
Status	설치 상태. 인증 완료(S_CERTIFICATED), 다운로드 완료(S_DOWNLOADED), 설치 종료(S_COMPLETED)의 중 하나



(a) 소프트웨어의 압축



(b) 전송 메시지 구성

(그림 4) 소프트웨어 전송시 zip 파일의 변형 및 암호 메시지

4.1.4 소프트웨어 압축

IMS가 관리하는 정품 소프트웨어들은 모두 ZIP 압축 유틸리티[11]로 압축되어 저장된다. 일반적으로 ZIP 형식은 (그림 4)(a)와 같이 파일 헤더와 데이터들의 리스트, 디렉토리 등으로 구성되어 있다. 디렉토리에는 모든 파일 헤더 정보를 요약하고 있으며 파일 이름, 압축 방법 등의 내용을

담고 있다. 따라서 각 파일 헤더와 디렉토리를 지운다면, 도청자가 압축 파일을 가지고 가더라도 이를 복원화 할 수 없게 된다. 따라서 본 논문에서는 Manager가 새로운 정품 소프트웨어가 도입되면 정품 소프트웨어를 ZIP으로 압축하고, ZIP 압축 파일 내부의 모든 파일 헤더와 디렉토리를 더미 데이터로 지운 뒤 변형된 이 ZIP 압축 파일과파일 헤더와 디렉토리의 원본 데이타를 따로 저장하여 관리한다.

4.1.5 소프트웨어 전송 시 메시지 구조

Server의 IMS는 설치 세션동안 압축된 소프트웨어 파일을 MAP으로 전송한다. 이를 위해 본 논문에서는 (그림 4)(b)와 같이 sw_list 테이블의 필드와 압축시 저장해 둔 원본 데이타(ZIP 파일 내의 파일 헤더와 디렉토리)로 전송 메시지를 구성하고 무결성을 위해 전송 메시지에 해쉬 값을 첨부하고 전송 메시지를 세션 키로 암호화하여 변형된 압축 소프트웨어와 함께 전송한다.

4.1.6 소프트웨어 등록 모듈

소프트웨어 등록 모듈은 새로운 소프트웨어가 도입될 때 압축 모듈을 호출하여 소프트웨어를 압축하여 디스크에 저장하고 DB 관리자를 통해 sw_list 테이블에 레코드를 추가한다.

4.1.7 세션 관리자

설치 세션과 보고 세션의 연결부터 종료까지 모든 작업을 처리한다. 세션 관리자에서 실행되는 알고리즘 ManageSession을 의사 코드로 기술하면 (그림 5)와 같다.

```
Algorithm ManageSession
Input : request from MAP
Output : SUCCESS or FAIL
{
    if(request from MAP is for installation session) {
        get the digital certificate from MAP ;
        certify it by calling 인증 모듈 ;
        if(certification fails)
            return FAIL ;
        set the status field of the installing_list table to
            S_CERTIFICATED ;
        generate a session key and encrypt it using the
            public key in the certificate by calling the
            암호화/복호화 모듈 ;
        send the encrypted session key to the MAP ;
        get information about the compressed software
        from DB 관리자 ;
        read the compressed software from disk drive ;
        build the transfer message shown in the Fig. 4(b)
            and encrypt using the session key by
            calling 암호화/복호화 모듈 ;
        send the encrypted transfer message to the MAP
            with the compressed software.
        set the status field of the installing_list table to
            S_DOWNLOADED ;
        get the result of installation of the software and
            update it to the installed_list table by calling the
            DB 관리자 ;
        set the status field of the installing_list table to
            S_COMPLETED ;
        return SUCCESS ;
    }
    else if(request from MAP is for report session) {
        get the digital certificate from MAP ;
        certify it by calling 인증 모듈 ;
        if(certification fails)
            return FAIL ;
        generate a session key and encrypt it using the
            public key in the certificate by calling 암호화/복
            호화 모듈 ;
        send the encrypted session key to the MAP;
        get the result of monitoring the PC registry from
            MAP, which is encrypted by the session key
            given to MAP ;
        decrypt the result with the session key by calling
            암호화/복호화 모듈 ;
        if(decryption fails) {
            send a fail message to the MAP ;
            return FAIL ;
        }
        insert the result into the report_list table ;
    }
    else
        return FAIL ;
}
```

(그림 5) 세션 관리자의 전체 흐름을 표현하는 의사 코드

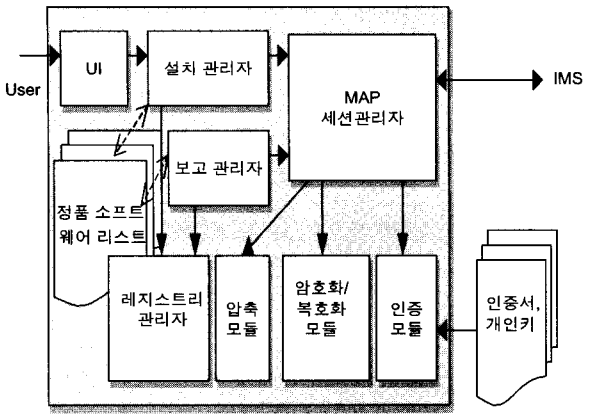
4.2 MAP 설계

4.2.1 MAP 구조 설계

MAP은 Client에서 실행되는 SAP 모듈로서 두 가지 기능을 수행한다. 첫째, Client 컴퓨터의 사용자인 User로부터 소프트웨어 설치 요구를 받아 IMS에 접속하여 소프트웨어를 설치한다. 둘째, 주기적으로 Client상에 설치된 소프트웨어 리스트를 조사하고 불법으로 설치된 소프트웨어를 IMS에 보고하는 기능을 수행한다. MAP은 (그림 6)과 같이 구성된다.

Client의 인증서와 개인키는 Server의 PCA로부터 인증서를 발급 받은 후 Client 컴퓨터의 시스템 파일 속에 기록된다. MAP은 자신이 설치한 정품 소프트웨어의 리스트를 가지고 있으며 이 리스트는 보안을 위해 디스크상에 저장하지 않고 메모리 공간에 저장 유지한다.

이 리스트의 복사본이 Server에도 역시 저장되어 있어 MAP 프로그램이 비정상적으로 종료하거나 리스트를 손실하였을 때 Server로부터 복구할 수 있다. 각 모듈에 대한 설명은 다음과 같다.



(그림 6) MAP 내부 구조

- 설치 관리자는 User의 지시에 의해 UI 모듈에 의해 호출되며 새로운 소프트웨어의 설치를 위해 설치 세션을 실행하는 세션 관리자 코드를 호출한다. 설치가 완료되면 정품 리스트에 추가한다.
- 보고 관리자는 주기적으로 호출되며 시스템 레지스트리를 검사하고 그 결과를 IMS에 통보하기 위해 보고 세션을 수행하는 세션 관리자 코드를 호출한다.
- 세션 관리자는 IMS에 접속하여 설치 세션과 보고 세션을 실행한다.
- 레지스트리 관리자는 시스템 레지스트리를 검색하여 불법 소프트웨어의 설치 여부를 검사한다.
- 압축 모듈은 세션 관리자로부터 호출되는 코드로서 설치 세션에 의해 IMS로부터 받은 소프트웨어의 압축을 푸는 기능을 제공한다.
- 암호화/복호화 모듈은 세션 동안 IMS와 주고 받는 메시지의 암호화와 복호화를 위한 코드를 제공한다.
- 인증 모듈은 인증 과정 동안 시스템에 설치된 인증서로부터 인증 ID 값을 읽거나, 개인키 값을 읽는 기능을 제공한다.

4.2.2 시스템 레지스트리(Registry) 검사

윈도우의 설치 프로그램(Install Shield)에 의해 설치된 모든 소프트웨어들은 윈도우의 시스템 레지스트리에 기록되며 윈도우에서 실행되는 정품 소프트웨어들은 윈도우 설치 프로그램인 Install Shield를 이용하여 설치되도록 제작되었다. MAP은 레지스트리를 검사하여 현재 설치된 소프트웨어 리스트를 확보하고 이를 자신이 설치한 정품 소프트웨어의 리스트와 비교하여 허락되지 않은 불법 소프트웨어를 판별한다.

4.2.3 설치 관리자

UI를 통해 사용자가 설치하는 메뉴를 실행하면 설치 관리자가 호출되어 (그림 7)과 같은 알고리즘 Install이 실행된다.

```
Algorithm Install
Input : nothing
Output : nothing
{
        send the certificate to IMS ;
        get the session key from IMS which is generated by
                IMS ;
        send a packet requesting the software list
                which is encoded with the session key ;
        get the installable software list from IMS ;
        make the User select a software to install on the Client ;
        send an encoded packet by the session key, which
                requests download of the software selected by
                User ;
        get the compressed software from IMS ;
        install the software automatically just after download ;
        send the packet of the installation result to IMS ;
        add the software name to the installed list managed by
                MAP ;
}
```

(그림 7) 설치 관리자에서 실행되는 알고리즘 Install의 의사 코드

4.2.4 보고 관리자

보고 관리자는 주기적으로 실행되는 코드로서 Client의 시스템 레지스트리를 검사하고 Client에 설치된 소프트웨어 리스트를 작성하고 정품 리스트와 비교하여 불법 소프트웨어의 설치 여부를 판단한다. 자신이 설치한 소프트웨어 리스트에 없는 소프트웨어를 발견하면 세션 관리자를 통해 보고 세션을 설정하고 결과를 IMS에 전송한다. 자신이 불법이라고 발견한 소프트웨어가 프리웨어나 쉐어웨어일 수 있기 때문에 IMS로부터 판단을 기다린다. 보고 관리자에서 실행되는 알고리즘 Report의 의사 코드는 (그림 8)과 같다.

```
Algorithm Report
Input : installed list
Output : TRUE or FALSE
{
        read all the installed software names from Windows
                Registry ;
        for(each software in the software names) {
                compare the software with the installed list ;
                if(the software is not found) { // it is an illegal
                        software.
                        make SessionManager send the result to
                                IMS by generating the report
                                session ;
                        get the result from SessionManager ;
                        if(the software is either freeware or
                                shareware)
                                return TRUE ;
                        else
                                return FALSE ;
                }
        }
        make SessionManager send the result to IMS, which
                generates the report session ;
        return TRUE ;
}
```

(그림 8) 보고 관리자에서 실행되는 알고리즘 Report의 의사 코드

5. SAP 시스템 구현

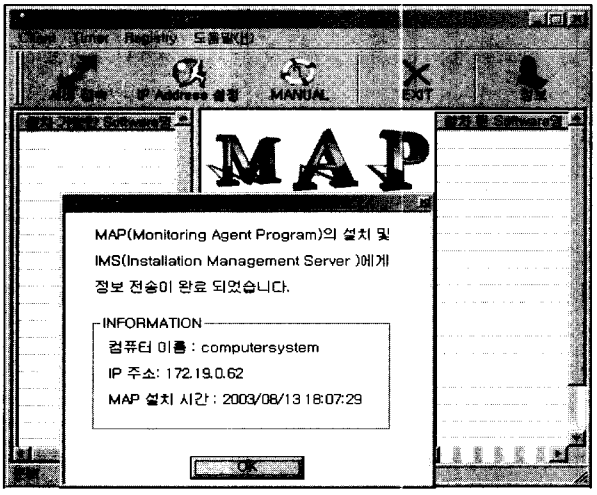
5.1 구현 시스템

본 장에서는 SAP 시스템의 구현에 대해 기술한다. 서버와 클라이언트간의 메시지 보안을 위해 마이크로소프트사의 CryptoAPI를 사용하였다[12, 13]. 이 API는 ASN.1에 정의된 인코딩/디코딩[14], 해쉬, 인증서 관리, 암호화, 복호화 등의 기능이 포함되어 있다.

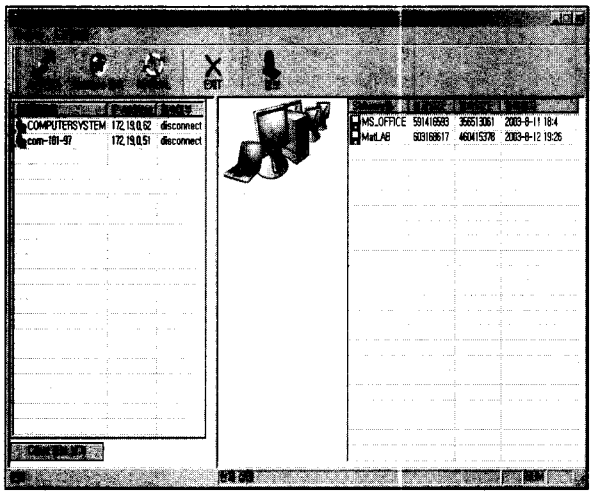
5.2 시스템 동작

5.2.1 MAP 프로그램 기동

아래 그림은 Client 컴퓨터에서 MAP을 설치한 후 초기 화면을 보여주고 있다. MAP 프로그램이 실행되면 Client 컴퓨터의 주요 사항들(컴퓨터 이름, IP 주소, MAP 설치 시간)을 IMS에 전송한다. 그리고 IMS는 Client 컴퓨터의 정보를 client_list 테이블에 등록한다. (그림 9)(a)는 MAP 프로그램이 Client에 처음 설치되어 기동하는 화면이며, (그림 9)(b)는 MAP이 설치되고 난 후 IMS에 MAP의 실체가 보이는 화면이다.



(a)

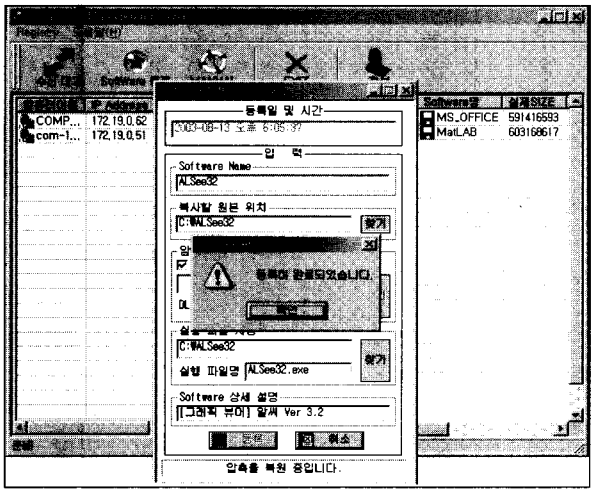


(b)

(그림 9) MAP이 처음 기동할 때의 MAP과 IMS의 상황

5.2.2 소프트웨어의 등록

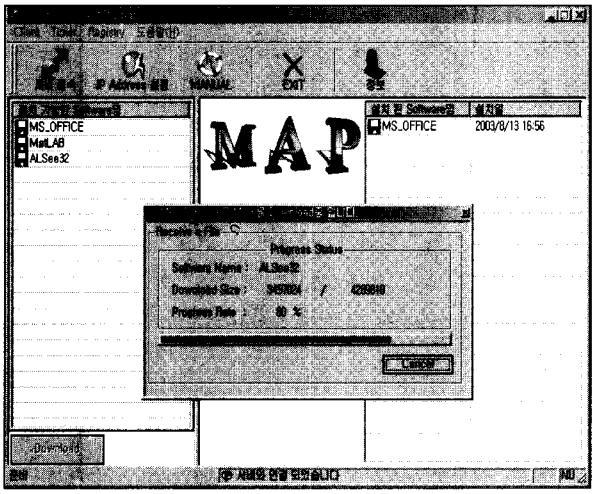
(그림 10)은 IMS가 ALSee32 소프트웨어를 압축 등록하는 과정을 보여 준다.



(그림 10) IMS에서 소프트웨어 등록 화면

5.2.3 소프트웨어 설치

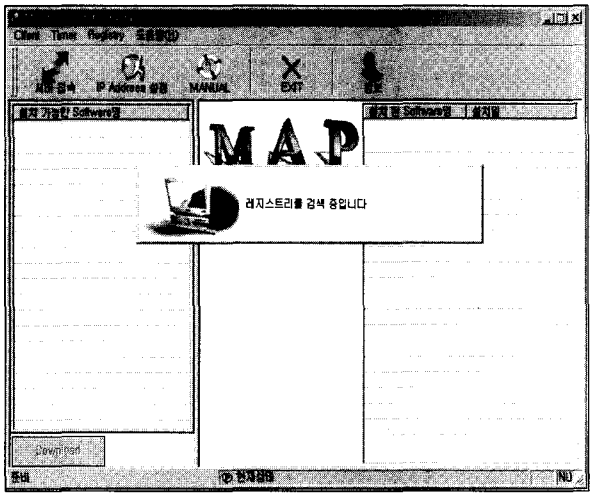
Client의 User가 MAP 프로그램을 이용하여 IMS 상에 존재하는 소프트웨어 목록을 가지고 와서 ALSee32 소프트웨어를 설치하는 과정은 (그림 11)과 같다. (그림 11)에서 왼쪽의 리스트는 IMS로부터 설치 가능한 소프트웨어 리스트를 보여주며 오른쪽은 MAP에 의해 설치된 소프트웨어를 보여 준다.



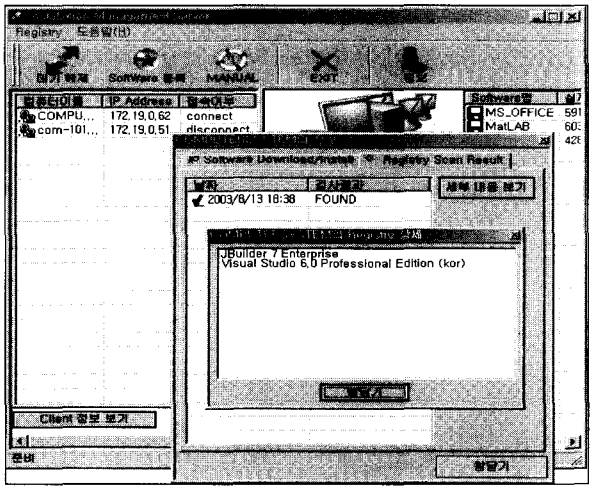
(그림 11) Client에서 Server로부터 소프트웨어 설치 화면

5.2.4 불법 소프트웨어 탐색

(그림 12)(a)는 MAP이 주기적으로 불법 소프트웨어를 탐색하는 과정으로서 현재 불법적으로 설치된 소프트웨어를 탐색하는 화면이다. 탐색 결과 Client에 불법적으로 설치된 JBuilder 7.0을 IMS에 자동으로 통보한다. (그림 12)(b)는 이 통보 결과를 받은 IMS의 상황을 보여준다.



(a)



(b)

(그림 12) Client에서 불법 소프트웨어 탐색 및 IMS의 결과 화면

6. 결 론

정보화 산업의 부작용으로서 소프트웨어 불법 복제에 따른 지적 재산권 침해가 증대하고 있다. 본 논문에서는 불법 소프트웨어의 설치 및 사용을 방지하는 소프트웨어 시스템을 제시하였다. 본 시스템에서는 두 가지 방법을 통해 불법 소프트웨어를 관리 및 통제한다. 첫째, 서버가 중앙에서 소프트웨어를 관리함으로써 소프트웨어의 유출 가능성이나 라이센스 관리의 효율성을 제공한다. 둘째, 클라이언트가 주기적으로 설치된 소프트웨어 리스트를 서버에게 전송하도록 함으로써, 서버가 쉽게 불법 소프트웨어의 설치 여부를 파악할 수 있는 기능을 제공한다.

한편, 클라이언트와 서버 컴퓨터간의 통신시 인증, 기밀성, 무결성의 보안 서비스를 제공함으로써 외부 세계로의 유출 및 변조를 방지하는 기능을 제공하고 있다. 그리고 이러한 기능을 실제 윈도우 환경에서 구현하여 시스템의 효율성 및 안전성을 검증하였다.

그러나 설치시 윈도우 레지스트리에 등록하지 않는 소프트웨어들은 발견할 수 없다는 문제점이 아직도 남아 있다. 향후에는 이 문제와 더불어 서버의 부하를 줄이기 위해서 효율적으로 소프트웨어를 디스크에 배치 관리하는 기법, 보안 프로토콜 등을 연구할 예정이다.

참 고 문 헌

[1] 월간 SW 저작권, SPC 집계 2002년 S/W불법복제 조사 리뷰, March, 2003년.

[2] 골든 시큐리티㈜, 홈페이지 : http://www.goldensecu.co.kr.

[3] Genesis㈜, 홈페이지 : http://www.genesis.co.kr.

[4] 체키㈜. 홈페이지 : http://www.checki.co.kr.

[5] 이미지튜브㈜, 홈페이지 : http://www.imagetube.co.kr.

[6] 박창섭, *암호 이론과 보안*, 대영사, 1999.

[7] S. Burnett and S. Paine, *RSA Security's Official Guide to CryptoGraphy, RSA Press*, 2001.

[8] R. Housley, W. Polk, W. Ford and D. Solo, Internet X.509 Public Key Infrastructure : Certificate and CRL Profile. RFC 2459, IETF, 1999.

[9] R. Rivest, The MD5 Message-Digest Algorithm, RFC 1321, 1992.

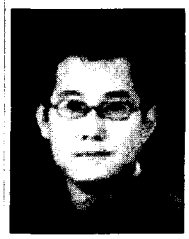
[10] FIPS 180-1, "Secure Hash Standards," Federal Information Processing Standards Publication, U.S. Department of Commerce/NIST, 1995.

[11] PKWare, ZIP File Format Specification, http://pkware.com/products/enterprise/white_papers/appnote.html, 2003.

[12] 강선명, Visual C++ 암호화 프로그래밍, 프리렉 출판사, 2003.

[13] Microsoft, MSDN Library, http://www.msdn.microsoft.com/library/default.asp, 2003.

[14] CCITT, Recommendation X.209 : Specification of Basic Encoding Rules for Abstract Syntax Notation One(ASN.1), 1988.



황 기 태

e-mail : calafk@hansung.ac.kr
1986년 서울대학교 컴퓨터공학과(학사)
1988년 서울대학교 대학원 컴퓨터공학과 (공학석사)
1994년 서울대학교 대학원 컴퓨터공학과 (공학박사)
2000년~2001년 University of California, Irvine 방문 교수
1994년~현재 한성대학교 컴퓨터공학부 부교수
관심분야 : 유비쿼터스 컴퓨팅, 인터넷 시스템, 모바일 보안 등

김 남 윤

e-mail : nykim@hansung.ac.kr
1992년 서울대학교 컴퓨터공학과(학사)
1994년 서울대학교 대학원 컴퓨터공학과 (공학석사)
2000년 서울대학교 대학원 컴퓨터공학과 (공학박사)
1999년~2002년 삼성전자 무선 사업부 책임 연구원
2002년~현재 한성대학교 정보공학부 전임강사
관심분야 : 이동통신 시스템, 정보 보안, 실시간 시스템 등