전자 상거래를 위한 소프트웨어 사용권 관리 에이전트 시스템

(Software License Management Agent System for Electronic Commerce)

윤 우 성 † 윤 정 모 †† 김 태 윤 †††
(Woo-Seong Yoon) (Jung-mo Yoon)(Tai-Yun Kim)

요 약 전자 상거래의 발전과 더불어 인터넷을 통한 소프트웨어의 분배 및 구매가 활성화되고 있다. 전자 소프트웨어 분배(ESD: Electronic Software Distribution) 모델중 선지불(Buy-first) 방법과 후지불(Try-before-buy) 방법은 소프트웨어의 불법 복제 문제를 해결하지 못한다. 최근 연구된 전자 사용권(EL: Electronic License) 모델은 소프트웨어와 사용권을 분리하여 불법 복제 문제를 해결한다. 그러나 이 방법 역시 소프트웨어에 대하여 다양한 지불 방법을 해결하지 못한다. 따라서 본 논문에서는 새로운 형태의 ESD 모델인 소프트웨어 사용권 관리 에이전트 시스템을 제안한다. 제안된 시스템은 새로운 형식의 사용권(NL: New License)을 제안하여 사용자가 원하는 다양한 지불 방법을 해결한다. 또한 사용 내역서(SC: Software Charge)를 제안하여 사용자가 사용한 소프트웨어 값을 판매자가 받는 것을 보장한다. 제안된 시스템의 에이전트는 NL과 SC를 관리하여 불법 복제 방지 효과 및 시스템간의 확장성을 제공한다.

Abstract With the growth of the EC(Electronic Commerce), Buying and selling software through the internet are expanded. Among the ESD(Electronic Software Distribution) methods, Buy-first method and Try-before-buy method can not solve the illegal copy problem. Recently developed EL(Electronic License) model solve the illegal copy problem by separating the software and license. But this method also can not support various ways for payment. In this paper we propose the software license management system that is a newly form like ESD model. This system proposes NL(New License) to support various payment methods and SC(Software Charge) to insure that a seller takes the software price. Agent of the proposed system offers scalability to other systems and illegal copy protection function by managing NL and SC.

1. 서 론

전자 상거래의 일반적인 정의는 네트워크와 컴퓨터 시스템을 통한 상품의 구매와 판매라고 할 수 있다[1,2]. 상거래에 필요한 정보와 처리 절차가 컴퓨터를 통해 이루어지므로, 거래의 신속성, 정확성, 효율성을 얻을 수 있다. 또한 공간적 제약을 받지 않고 거래를 이룰 수 있으므로, 보다 많은 거래 대상에 접근하여 거래 성사의 가능성을 높일 수 있다. 특히 최근 전자상거래를 통한 소프트웨어 판매 규모가 괄목한 만한 성장을 이루고 있다[1,3]. 온라인을 통한 소프트웨어 판매 증가에 편승하여 사용자가 요구하는 소프트웨어의 사용 욕구는 일회성에서부터 영구적인 사용까지 다양하다[3].

인터넷을 통한 ESD 모델에는 선지불 방법, 후지불 방법 그리고 EL 모델이 있다[4,5,6]. 선지불 방법과 후지불 방법은 판매자가 일단 소프트웨어를 판매하고 나면, 소프트웨어의 불법 복제 및 유통을 방지하기 어렵다. 또한 선지불 방법과 후지불 방법은 사용자의 지불 시기에 따라서 분류되는 방법으로서 사용자가 요구하는

† 비 회 원 : 고려대학교 컴퓨터학과
wsyoon@netlab.korea.ac.kr
†† 종신회원 : 서울산업대학교 전자계산학과 교수
jmyoon@plaza1.snut.ac.kr
††† 종신회원 : 고려대학교 컴퓨터학과 교수
tykim@netlab.korea.ac.kr
논문접수 : 2000년 5월 4일
심사완료 : 2000년 11월 6일

다양한 형태의 지불 요건을 만족하지 못한다[6]. 최근 제시된 EL 모델은 소프트웨어와 사용권을 분리하여 관리하므로 소프트웨어 판매 후 사용자의 불법 복제 및 유통을 방지하는 효과가 있다[4,5]. 그러나 이 방법 역시 사용자가 요구하는 다양한 지불 방법을 해결하지는 못한다. 따라서 사용자가 원하는 다양한 지불 방법을 충족하는 동시에 소프트웨어의 불법 복제 및 유통을 방지하는 새로운 방법이 필요하다.

본 논문은 공개키 기반 구조(PKI: Public Key Infrastructure)[7,8]에 기초하여 새로운 형식의 사용권(NL: New License)을 제안한다. NL은 다양한 형태의 지불 방법을 제공할 수 있다. NL 관리는 구매자 시스템(BS: Buyer System)의 사용권 관리 에이전트(LMA: License Management Agent)를 통하여 이루어진다. 또한 사용 내역서(SC: Software Charge)를 제안한다. SC는 소프트웨어를 사용한 시간을 기록하여 판매자에게 사용자가 사용한 소프트웨어에 대한 정당한 값을 지불하는 것을 보장한다. SC는 판매자 시스템(SS: Seller System)의 사용자 관리 에이전트(UMA: User Management Agent)가 관리한다. 제안된 LMA와 UMA는 암호 프로토콜을 이용하여 다른 판매자와 사용자간의 인증 및 확장성을 제공한다. 네트워크 상에서 LMA와 UMA 사이의 통신은 atomicity[9,10]한 방법을 사용한다.

본 논문의 구성은 다음과 같다. 2장에서 현재 사용되고 있는 ESD 모델들에 관하여 소개하고 문제점을 제시한다. 3장에서 다양한 지불 방법을 해결하기 위한 NL과 SC를 제안하고, 4장에서 소프트웨어 사용권 관리 에이전트 시스템 구현을 설명한다. 5장에서 본 시스템의 구현 결과 및 성능을 평가한다. 6장에서 결론 및 향후 연구 방향을 제시한다.

2. ESD 모델

본 장에서는 현재까지 진행되어온 ESD 모델을 소개한다. 동시에 기존의 ESD 모델들을 분석하여 보고 문제점을 지적한다. 연구 결과를 토대로 다양한 지불 방식을 지원하면서, 불법 복제를 막을 수 있는 새로운 ESD 모델을 제안하고자 한다.

2.1 선지불(Buy-first) 방법

선지불 방식은 사용자가 상품을 선택하고 상품의 가격을 먼저 지불한 후 소프트웨어를 설치하는 방법이다. 선지불 방법의 경우 판매자는 소프트웨어를 암호화하고 암호화한 소프트웨어를 패키지화 한다. 이렇게 패키지화한 소프트웨어를 BOB(Bag of Bits)[6]라고 한다. 사용자는 BOB를 다운로드하고, 상품의 가격을 지불한다. 판매자는 사용자가 지불을 완료하면 키를 전달한다. 사용자는 전달 받은 키를 이용하여 BOB를 복호화하고 소프트웨어를 설치한다. 선지불 방법을 적용한 소프트웨어 분배 시스템은 포틀랜드 소프트웨어(Portland Software)사의 집락(ZipLock) 시스템을 예로 들 수 있다[11]. 집락 시스템은 대부분의 판매 시스템에서처럼 구매자의 전자 메일을 통하여 키를 전달한다[5,6]. 그림 1은 선지불 방법을 이용한 소프트웨어 구매 절차이다. 선지불 방법은 구매 절차가 간단하다. 그러나 사용자는 소프트웨어를 구매하기 전에는 소프트웨어에 대한 자신의 만족도를 평가할 수 없으며 네트워크 상에서 BOB와 키를 다른 이에게 노출 당하거나 사용자가 악의를 품는 경우 불법 복제 및 유통을 막을 수 없다.

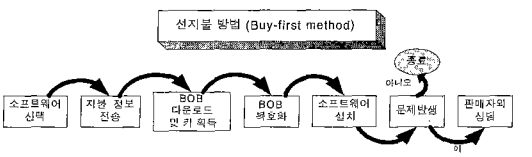

그림 1 선지불 방법을 이용한 소프트웨어 구매 절차

2.2 후지불(Try-before-buy) 방법

후지불 방법은 구입하고자 하는 소프트웨어를 사용한 후 구입 여부를 결정하는 시스템이다. 판매자는 날짜에 기초로 하여 사용기간에 제한을 둔 소프트웨어 또는 제공하는 기능에 제한을 둔 시범 소프트웨어를 제공한다[6]. 사용자가 소프트웨어를 설치하면 제한된 사용 기간 동안 사용할 수 있다. 제한된 사용 기간이 지나게 되면, 설치된 소프트웨어는 자동으로 삭제된다. 사용자가 구매를 원하는 경우 온라인을 통하여 구매의사를 밝힌다. 사용자가 상품의 값을 지불하고, 판매자는 사용자 시스템의 잠금(lock) 장치를 해제한다[6]. 후지불 방법을 적용한 소프트웨어 분배 시스템은 포틀랜드 소프트웨어사의 브이박스(Vbox) 시스템이 있다[11]. 그림 2는 후지불 방법을 이용한 소프트웨어 구매 절차를 나타낸다. 후지불 방법을 이용한 소프트웨어 판매 방식은 선지불 방법

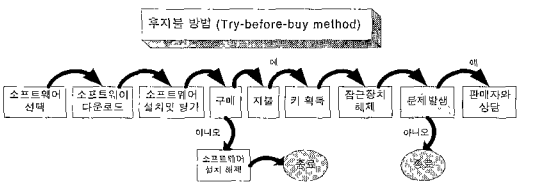

그림 2 후지불 방법을 이용한 소프트웨어 구매 절차

보다 사용자의 만족이 높다. 그러나 구매 절차가 복잡해지고, 잠금 장치를 해제한 소프트웨어의 불법 복제 및 유통을 방지할 수 없다.

2.3 전자 사용권(EL: Electronic License) 모델

EL 모델은 소프트웨어의 사용권을 제품으로부터 분리시킨 후 사용권을 관리하는 시스템이다. EL 모델에서 사용자는 원하는 소프트웨어를 즉시 다운로드 할 수 있다. 하지만 어떤 소프트웨어 제품이 PC상에 설치되어 있더라도 사용권이 없으면 수행되지 않기 때문에 소프트웨어의 사용을 위해서는 지불과 등록을 통하여 사용권을 전달받아야 한다. 이 시스템은 소프트웨어가 실행될 때 현재 사용자가 제품의 사용권이 있는지 확인하고 확인 결과에 따라 소프트웨어가 계속 작동하던가 작동이 중지되거나 하기 때문이다[12]. 시멘텍사(Symantec)에서는 EL 모델을 적용한 소프트웨어의 온라인 판매가 이루어지고 있다[13]. 그림 3은 EL 모델을 이용한 소프트웨어 구매 절차이다. EL 모델은 사용권을 따로 관리하므로 불법 복제 및 유통을 방지하는데 효과적이다. 그러나 다양한 종류의 소프트웨어에 대하여 사용자가 원하는 적절한 지불 방법을 제공하지는 못한다.

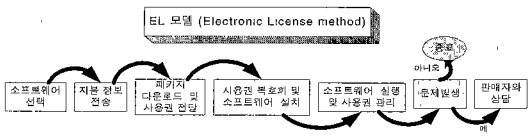

그림 3 EL 모델을 이용한 소프트웨어 구매 절차

3. 제안한 사용권(NL)과 사용 내역서(SC)

기존의 ESD 모델은 사용자가 원하는 다양한 지불 방법을 제공하지 못한다. 이를 해결하기 위하여 새로운 사용권(NL)을 제안한다. 또한 지불 처리를 보장하기 위하여 사용 내역서(SC)를 제안한다.

3.1 NL의 형식

사용자가 요구하는 다양한 지불 방법을 판매자가 지원하기 위한 NL의 형식은 그림 4와 같다.

NL의 최초 256비트는 소프트웨어 제품의 일련 번호(S_ID: Software ID)이다. 소프트웨어 일련 번호는 유일한 값으로서 소프트웨어의 정보를 관리하기 위함이다. 다음의 128비트는 사용자가 판매자를 인증하기 위하여 판매자 ID(D_ID: Dealer ID)를 MD5 해쉬 알고리즘을 이용하여 메시지 다이제스트[14]한 값이다. 이것은 사용자가 판매자를 인증하기 위함이다. 사용권 정보는 사용자가 구매한 소프트웨어를 사용할 수 있는 시간 정보를

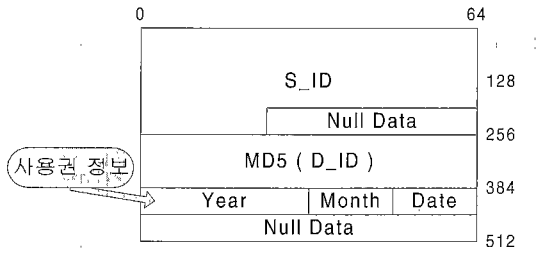

그림 4 새로운 사용권(NL)의 형식

담고 있다. 사용권 정보에는 세 가지가 있다. 첫 번째 사용자가 원하는 사용일수를 사용권 권한 정보에 기록하는 것이다. 두 번째 사용자가 소프트웨어의 영구적인 사용권을 원하는 경우 모든 값을 9로 설정한다. 세 번째 사용자가 소프트웨어를 설치한 후 소프트웨어 사용 시간에 따른 지불 방법을 원하는 경우 사용권 권한 정보의 모든 값을 0으로 설정한다.

3.2 SC의 형식

소프트웨어의 사용 시간 정보를 판매자에게 전달하기 위한 SC의 형식은 그림 5와 같다.

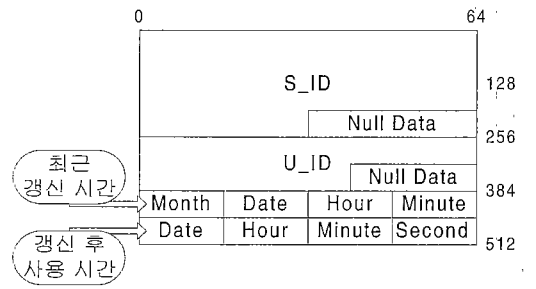

그림 5 제안한 사용 내역서(SC)의 형식

SC의 최초 256비트는 S_ID이다. 판매자에게 사용자가 사용한 소프트웨어 정보를 알리기 위함이다. 다음 128 비트는 사용자 ID(U_ID: User ID)이다. U_ID는 현재 소프트웨어를 사용하고 있는 사용자가 정당한 사용자임을 판매자에게 확인 받기 위함이다. 다음의 64비트는 SC를 판매자에게 전달하여 소프트웨어 값을 지불한 최근 갱신 시간 정보를 기록한다. 다음의 64비트는 SC를 갱신한 후에 사용자가 소프트웨어를 사용한 시간을 누적한 것이다.

4. 소프트웨어 사용권 관리 에이전트 시스템 구현

소프트웨어 사용권 관리 에이전트 시스템은 기존의 ESD 모델을 기초로 하여 구현되었다. 본 논문에서 제안한 시스템은 사용자가 원하는 다양한 지불 방법과 네트워크상의 다른 주체간의 연결이 가능한 확장성을 제공하기 위하여 에이전트를 사용한다.

4.1 소프트웨어 사용권 관리 에이전트 시스템의 구성

소프트웨어 사용권 관리 에이전트 시스템에 참가하는 주체로서는 사용자 시스템(BS), 사용권 관리 에이전트(LMA), 판매자 시스템(SS) 그리고 사용자 관리 에이전트(UMA)가 있다. Payment Gateway(PG)는 금융 기관과의 연결을 나타낸다. 각각의 SS는 PG를 통하여 금융 기관과 연결되어 있어서 사용자에게 다양한 지불 서비스를 제공할 수 있다고 가정한다. 그림 6은 소프트웨어 사용권 관리 에이전트 시스템의 전체 그림이다. 소프트웨어 구매는 BS가 SS들로부터 소프트웨어 패키지(SP)를 다운로드 받고, 새로운 사용권(NL)을 전달받는 절차이다. 사용권 관리는 SP와 NL을 이용하여 소프트웨어를 설치 또는 실행할 때 BS와 LMA사이에 이루어지는 소프트웨어 사용 허가에 관한 수행 절차이다. 사용 내역서 전달 프로토콜은 LMA가 BS의 사용 내역서(SC)를 UMA에게 전달하는 절차이다. 지불 처리는 SS에서 UMA가 전달받은 SC를 조회하여 소프트웨어 지불을 처리하는 절차이다.

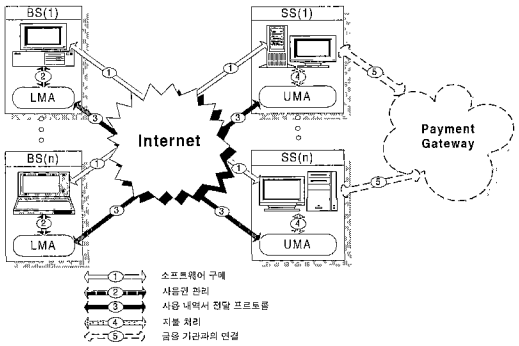

그림 6 소프트웨어 사용권 관리 에이전트 시스템

4.2 소프트웨어 구매

소프트웨어 사용권 관리 에이전트 시스템에서 첫 번째 수행되는 과정은 사용자가 SS에서 SP와 NL을 받는 것이다. 사용자는 판매자의 홈페이지를 통하여 원하는 소프트웨어를 선택할 수 있다. 판매자는 먼저 사용자를 인증하여야 한다. 판매자가 사용자들을 회원으로서 관리하면 소프트웨어 구매시 별도의 인증 절차를 거치지 않아도 된다. 인증 받은 사용자는 판매자가 제작한 SP를 다운로드 할 수 있다. 판매자는 사용자가 원하는 지불 방법에 따르는 상품의 값을 알려준다. 사용자는 기존의 지불 시스템을 통하여 상품의 값을 지불한다. 사용자가 사용 시간에 따른 지불 방법을 원하는 경우라면, 판매자는 사용자의 별도의 지불 없이 NL을 생성한다.

본 시스템에서는 판매자가 사용권을 암호화하여 사용자에게 전자 메일로 전달한다. 공개키 기반 구조를 가정하므로 판매자는 사용자 공개키(P_{user})와 판매자 비밀키(S_{seller})를 가지고 있다. 사용자도 판매자 공개키(P_{seller})와 사용자 비밀키(S_{user})를 가지고 있다. SS는 생성한 NL을 BS만 사용할 수 있게 하기 위하여 P_{user}로 암호화한다. SS는 이것을 다시 S_{seller}를 이용하여 전자 서명한다. SS가 BS에게 보내는 암호화된 NL은 다음과 같다.

$$\text{암호화된 NL} = S_{seller}(P_{user}(NL))$$

4.3 사용권 관리

4.3.1 LMA와 소프트웨어의 설치

BS에 설치된 LMA는 소프트웨어 설치시 NL을 이용하여 사용자와 소프트웨어를 인증한다. 사용자는 자신의 비밀키를 이용하여 NL을 획득한 다음 NL을 조작할 수 있다. 그러나 사용자는 판매자의 비밀키를 모르기 때문에 암호화된 NL을 생성할 수 없다. 사용자가 NL을 조작하는 것을 방지하기 위하여 LMA는 암호화된 NL을 전달받는다. LMA는 BS로부터 사용자 비밀키를 전달받는다. 사용자의 비밀키는 절대 외부에 공개되어서는 안되기 때문에, LMA는 BS에서 데몬(Daemon) 프로세스로 실행되어야 한다. BS는 루프백 어드레스(127.0.0.1)를 이용하여 LMA에게 데이타를 전달한다.

사용자는 LMA이 설치시 자신의 ID를 입력한다. 따라서 LMA는 사용자 ID, U_ID_{LMA}를 가지고 있다. SP는 사용자가 입력한 U_ID_{SP}와 SP 생성시 기록된 S_ID_{SP}를 LMA에게 전달한다. LMA는 BS로부터 $S_{seller}(P_{user}(NL))$ 양식을 지니는 암호화된 사용권을 입력

표 1 소프트웨어 설치 절차에 사용되는 알고리즘

알고리즘	설명
P_{user}	사용자의 공개키를 이용하여 평문을 암호문으로 암호화 한다.
S_{seller}	판매자의 비밀키를 이용하여 평문을 암호문으로 암호화 한다.
P_{seller}	판매자의 공개키를 이용하여 암호문을 평문으로 복호화 한다.
S_{user}	사용자의 비밀키를 이용하여 암호문을 평문으로 복호화 한다.
Get_{S_ID}	주어진 메시지에서 S_ID를 읽는다.

표 2 소프트웨어 설치 절차에 사용되는 데이타 요소

데이타 요소	설명
NL	제안한 새로운 형식의 사용권
U_ID_{LMA}	LMA 설치시 사용자가 입력한 사용자 ID
U_ID_{SP}	SP 설치시 사용자가 입력한 사용자 ID
S_ID_{SP}	SP가 가지고 있는 소프트웨어 ID
S_ID_{SS}	SS 시스템에서 NL 생성시 입력한 소프트웨어 ID

받는다. LMA는 $S_{seller}(P_{user}(NL))$을 P_{seller}로 복호화하여 SS가 보낸 것임을 확인한다. LMA는 이것을 S_{user}로 복호화하여 NL을 얻는다. LMA는 NL에서 SS가 발행한 S_ID_{SS}를 얻는다. SP가 전송한 U_ID_{SP}와 LMA에 기록되어 있는 U_ID_{LMA}를 비교하여 사용자를 확인한다. S_ID_{SP}와 S_ID_{SS}를 비교하여 소프트웨어 정품을 확인한다. LMA는 일련의 과정을 제대로 마치면 OK 메시지를 잘못되는 경우 NOK 메시지를 전달한다.

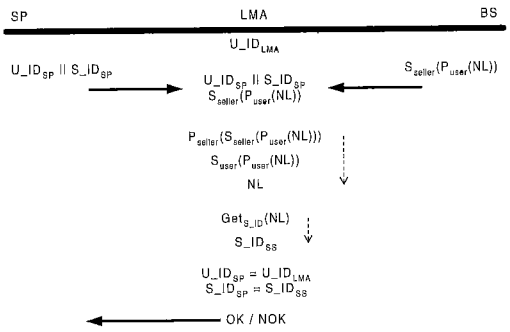

그림 7 소프트웨어 설치 절차

4.3.2 소프트웨어 실행과 사용권 관리

LMA는 소프트웨어 실행시 NL 관리 및 SC를 작성한다. 그림 8은 소프트웨어 실행시 LMA가 NL을 관리하는 것을 나타내었다.

LMA는 다음과 같은 일을 수행한다.

① NL 관리 쓰레드(thread) : 소프트웨어 정품 인증 및 NL을 관리한다.

② 내부 시간 관리 쓰레드 : 사용자가 사용시간에 따른 지불 방법을 원하는 경우 시간 관리가 중요하다.

③ 네트워크 감시 쓰레드 : 사용자가 소프트웨어를 사용한 시간 정보 등을 판매자 에이전트에게 전달하는 역할을 한다.

NL 관리 절차는 다음과 같다.

① 소프트웨어가 실행되는 순간 소프트웨어는 LMA에게 S_ID_{SP}를 전달한다.

② LMA는 전달받은 S_ID_{SP}와 일치하는 NL을 읽어온다. S_ID_{SP}와 일치하는 NL이 없는 경우 소프트웨어의 정품을 인증 할 수 없다.

③ LMA는 NL의 사용권 정보를 판단하여 유효한 경우 다음 단계로 넘어간다. 그렇지 않은 경우 프로세서를 강제 종료한다.

④ 사용자가 사용 시간에 따른 지불 방법을 선택한 경우 LMA는 사용자의 소프트웨어 사용 시간을 기록할 필요가 있다.

⑤ 내부 시간 관리 쓰레드가 현재 수행 중인 소프트웨어의 사용시간 관리를 한다.

⑥ 네트워크 감시 쓰레드가 소프트웨어 사용 정보를 담고 있는 SC를 제작하여 전송한다.

⑦ 소프트웨어 사용을 마치면 소프트웨어의 NL 관리가 끝나게 된다. 프로세서 ID가 확인되지 않는 순간 LMA는 소프트웨어의 사용이 끝난 것을 간주한다.

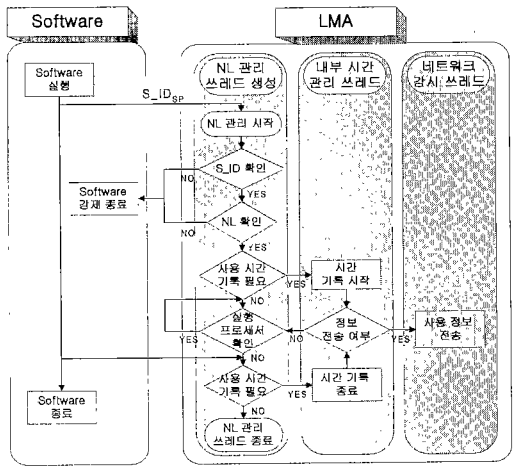

그림 8 소프트웨어 정품 인증 및 NL 관리 흐름도

4.4 SC 전달 프로토콜

UMA는 SS에 설치되어서 LMA부터 SC를 전달받는 에이전트이다. UMA가 LMA로부터 SC를 전달받는 과정은 그림 9와 같다.

LMA는 NL로부터 $MD5(D_ID_{LMA})$를 가지고 있다. UMA는 설치시 D_ID_{UMA}를 입력받아서 $MD5(D_ID_{UMA})$를 생성한다. 먼저 LMA는 SC가 제 3자에 의해 변형 및 변조되는 것을 막기 위하여 판매자 공개

표 3 SC 전달 절차에 사용되는 알고리즘

알고리즘	설명
MD5	MD5 해쉬함수를 이용하여 메시지를 암호문으로 만든다.
P_{seller}	판매자의 공개키를 이용하여 평문을 암호문으로 암호화 한다.
S_{user}	사용자의 비밀키를 이용하여 평문을 암호문으로 암호화 한다.
P_{user}	사용자의 공개키를 이용하여 암호문을 평문으로 복호화 한다.
S_{seller}	판매자의 비밀키를 이용하여 암호문을 평문문으로 암호화 한다.

표 4 SC 전달 절차에 사용되는 데이타 요소

데이타 요소	설명
D_ID_{LMA}	SS가 생성한 NL에 저장되어 있는 판매자 ID
D_ID_{UMA}	UMA 설치시 판매자가 입력한 판매자 ID
SC	사용 내역서
UIP	BS의 인터넷 주소(User IP Address)

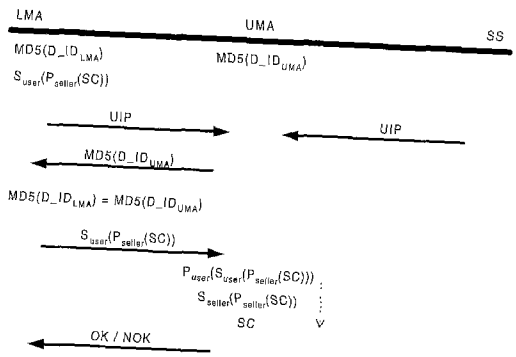

그림 9 SC 전달 절차 프로토콜

키 P_{seller}로 암호화한다. 판매자의 공개키로 암호화된 $P_{seller}(SC)$를 LMA는 사용자의 비밀키 S_{user}로 전자 서명한다. LMA는 이렇게 암호화된 $S_{user}(P_{seller}(SC))$를 전달하기 위한 절차로 UIP를 UMA에 전송한다. SS에게 SC가 필요한 경우 저장하고 있는 UIP를 전달한다. UMA는 전달받은 UIP로 $MD5(D_ID_{UMA})$를 전송한다. LMA는 전송 받은 $MD5(D_ID_{UMA})$와 NL에서 얻은 $MD5(D_ID_{LMA})$를 비교하여 판매자를 인증 한다. LMA는 $S_{user}(P_{seller}(SC))$를 UMA에 전송한다. UMA는 $S_{user}(P_{seller}(SC))$를 복호화하여 SC를 얻는다. UMA가 SC를 처리하고 나면 결과로서 OK 메시지를 보낸다. LMA는 UMA로부터 OK가 올 때까지 기다린다. 응답이 오면 LMA는 SC를 초기화한다. 일정 시간 후 응답이 오지 않는 경우 재전송 한다. LMA와 UMA 사이의 프로토콜은 atomicity[9,10]한 방법 적용하였기 때문에 응답을 받지 못하면 처음 상태로 복구한다. 이것은 사용자와 판매자간의 데이타가 동일하게 유지되는 것을 보장한다.

4.5 지불 처리

판매자가 지불 처리하는 시점은 SS가 지불 처리를 필요로 하는 경우와 사용자가 SS에 지불 처리를 요청하는 경우이다. 판매자가 지불 처리를 필요로 하는 경우 U_ID와 저장하고 있는 UIP를 UMA에 전달한다. UMA는 사용자의 최신 정보를 조사한다. 24시간 이내에 갱신된 자료가 없는 경우, 전달받은 UIP를 이용하여 LMA와 연결하여 SC를 전달받는다. 그림 10은 지불 처리 절차를 나타내었다.

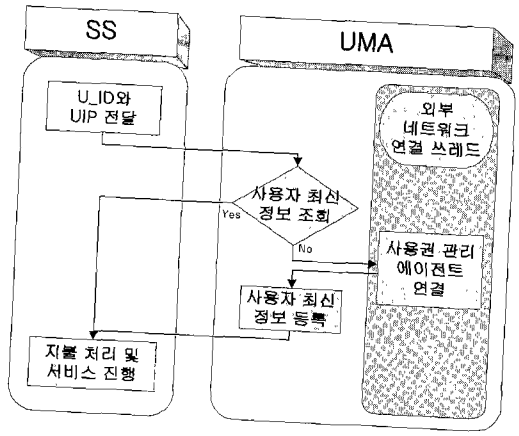

그림 10 지불 처리 절차 흐름도

판매자는 UMA에 저장되어 있는 SC의 사용 시간 정보와 S_ID를 읽어드린다. SC로부터 읽어 드린 소프트웨어를 사용한 시간과 S_ID로부터 찾아낸 소프트웨어의 단위 시간당 가격을 곱하여 값을 계산해 낸다. 판매자는 지불 수단을 통하여 이것을 처리한다.

5. 시스템 구현 결과 및 성능 평가

본 장에서는 제안한 소프트웨어 사용권 관리 에이전트 시스템의 상세 설계, 구현 환경 그리고 구현 결과를 제시한다. 또한 기존의 ESD 모델을 적용한 시스템과

본 논문에서 제안한 시스템과 비교한다.

5.1 시스템 상세 설계

본 논문에서 제안한 시스템은 사용자가 원하는 소프트웨어와 지불 방법을 결정하여 이를 사용자 시스템에 설치하는 과정과 설치한 소프트웨어를 사용하는 두 가지 단계로 나눌 수 있다. 그림 11은 제안한 시스템에서 발생하는 일련의 과정을 상세 설계한 그림이다.

① 사용자는 판매자 홈페이지에서 사용자 인증을 확인 받고, 사용자 ID를 발급 받는다. 소프트웨어 관리에 필요한 LMA를 다운로드 받는다.

② 사용자 ID를 입력하기만 하면 LMA 설치는 완료된다. LMA는 소프트웨어 ID를 이용하여 각각의 소프트웨어를 관리하므로 하나의 LMA만 사용자 시스템에 설치하면 된다.

③ 사용자는 원하는 상품을 선택하고 사용권 제작에 필요한 정보를 입력한다. 사용자는 홈페이지를 통하여 소프트웨어 패키지(SP)를 다운로드하고, 암호화된 사용권을 전자 메일로 전달받는다.

④ 소프트웨어를 설치하기 위하여 사용자 ID를 입력한다. 이 사용자 ID는 ⑥번 과정에서 LAM가 가지고 있는 사용자 ID와 비교함으로서 소프트웨어를 설치하려는 사람이 정당한 사용자임을 LMA로부터 인증 받는데 사용된다.

⑤ SP는 사용자로부터 전달받은 사용자 ID와 SP 생성시 판매자가 기록한 소프트웨어 ID를 LMA에게 전달한다. 이 소프트웨어 ID는 ⑥번 과정에서 LMA가 NL로부터 가져온 소프트웨어 ID를 비교하여 소프트웨어를 인증한다.

⑥ LMA는 사용자가 입력한 암호화된 사용권을 복호화하고 사용자 ID와 소프트웨어 ID를 이용하여 사용자가 SP를 인증한다. 이 과정은 4.3.1 LMA와 소프트웨어의 설치에서 자세히 설명하였다.

⑦ LMA는 인증 결과를 SP에게 전달하고 SP는 인증 결과가 OK이면 소프트웨어 설치를 수행한다.

⑧ 여기서부터 소프트웨어가 수행되는 과정을 설명한다. 소프트웨어는 LMA에게 소프트웨어 ID를 전달한다.

⑨ LMA는 전달받은 소프트웨어 ID와 암호를 푼 사용권을 이용하여 소프트웨어의 사용권을 인증하고 소프트웨어 관리를 시작한다. 이 과정은 본문 4.3.2 소프트웨어 실행과 관리에서 자세히 다루었다.

⑩ 사용자가 소프트웨어를 종료한 것이 확인되면, LMA는 SC를 제작하고 이를 암호화한다.

⑪ LMA는 UMA에게 자신의 주소를 알려준다. UMA는 LMA에게 MD5(판매자 ID)를 전달하여 판매

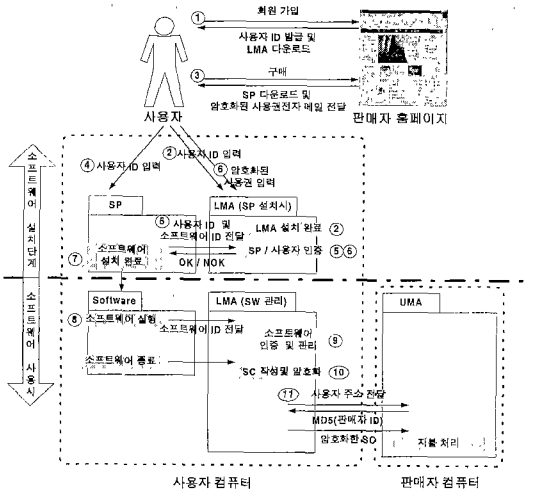

그림 11 소프트웨어 사용권 관리 에이전트 시스템 상세 설계

자임을 인증 받는다. LMA는 이를 확인한 후 암호화한 SC를 판매자 시스템의 UMA에 전달한다. 이 과정은 본문 4.4 SC 전달 프로토콜에서 상세히 다루었다.

5.2 시스템 구현 결과

본 논문에서 제안하는 소프트웨어 지불 관리 에이전트 시스템의 BS는 Windows98 운영체제와 Pentium MMX200㎒ CPU 환경에서 자바 JDK1.2 버전을 이용하여 구현하였다. SS는 동일한 환경에서 자바 웹 서버(JWS : Java Web Server)와 자바 서블릿을 이용하여 구현하였다. 시스템에서 사용되는 암호 알고리즘은 자바 암호 라이브러리 IAIK2.51[15] 버전을 사용한다. NL의 MD5(D_ID)는 메시지 다이제스트 알고리즘 중 MD5[15]를 사용한다. NL과 SC의 전송에 사용되는 공개키 암호 알고리즘은 RSA[14,16,17]를 사용한다.

LMA와 UMA는 시스템에서 데몬(Daemon) 프로세서로 실행되므로 별도의 사용자 인터페이스를 제공하지 않는다. 여기서는 소프트웨어 설치 과정만 제시한다. 그림 12는 소프트웨어 설치시 소프트웨어 패키지(SP)에 사용자 ID(U_ID_{SP}), 비밀 번호, 주민 등록 번호를 입력하는 과정이다. 사용자는 SP를 다운로드하고, SP는 일종의 인스톨 파일로서 SP를 이용하여 소프트웨어를 설치한다. 따라서 SP를 실행하려면 LMA에게 사용자 인증을 받아야 한다. 이때 SP는 그림 12에서 보듯이 사용자 ID(U_ID_{SP}), 비밀 번호, 주민 등록 번호를 LMA에게 전달하여 정당한 사용자임을 인증 받는다. 동시에 SP가 가지고 있는 소프트웨어 ID를 전달하여 소프트웨어 또

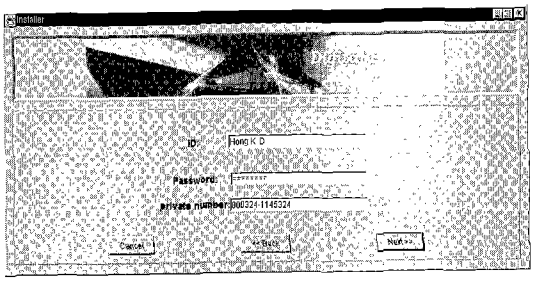

그림 12 사용자 정보 입력 화면

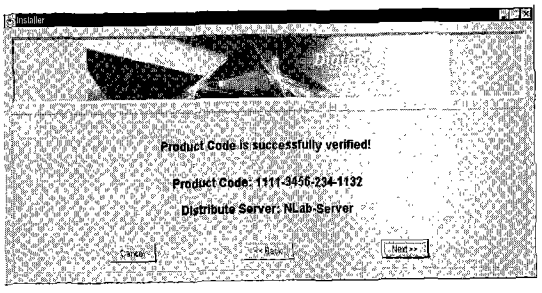

그림 13 소프트웨어 인증 화면

한 인증 받는다. LMA가 인증을 마친후 OK 신호를 SP에게 전달하면 소프트웨어 설치가 시작된다. 그림 13은 사용자 ID(U_ID$_{SP}$)와 소프트웨어 ID(S_ID$_{SP}$)를 인증받은 후 설치가 시작되는 소프트웨어 인증 화면이다.

5.3 성능 평가

표 5는 선지불 방법을 이용하는 집락 시스템(Zip), 후지불 방법을 이용하는 브이박스 시스템(Vbox), 시멘텍사의 판매 시스템(Sym) 그리고 본 논문에서 제안한 소프트웨어 사용권 관리 에이전트 시스템(Ns)에서의 소프트웨어 분배 및 구매 절차에 사용되는 모듈을 비교한 것이다.

분배 절차(D)에 있어서 ESD 모델을 이용하는 방법들은 모두 패키지를 제작한다. Zip은 패키지자체를 암호화하는 반면에 Sym과 Ns는 사용권을 따로 두어 사용권만 암호화한다. 따라서 Zip에서는 소프트웨어의 파일 크기가 큰 경우 암호화하는데 오랜 시간이 소요되는 단점이 있다.

구매 절차(P)에 있어서 Zip은 분배 절차와 같은 시간적인 오버헤드가 있다. Vbox는 소프트웨어의 잠금 장치를 직접 해제하는 반면에 Sym과 Ns는 사용권을 복호화하여 패키지의 잠근 장치를 해제한다. 따라서 Vbox에서는 잠금 장치 해제에 따르는 추가적인 절차가 필요하다.

관리절차(M)에 있어서 Zip은 소프트웨어 판매 후 사

표 5 시스템간의 분배 및 구매 절차에 따른 수행 모듈 비교

단계	적용 모듈	집락 시스템 (Zip)	브이박스 시스템 (Vbox)	시멘텍사의 판매 시스템 (Sym)	제안한 시스템 (Ns)
분배 절차 (D)	패키지의 제작	○	○	○	○
	패키지의 암호화	○	×	×	×
	사용권 제작	×	×	○	○
	사용권 암호화	×	×	○	○
구매 절차 (P)	패키지의 복호화	○	×	×	×
	패키지의 잠금 해제	○	×	○	○
	소프트웨어 잠금 해제	×	○	×	×
	사용권 복호화	×	×	○	○
관리 절차 (M)	소프트웨어 설치 해제	×	○	○	×
	사용권 관리	×	×	○	○

(○: Yes ×: No)

표 6 제안한 소프트웨어 사용권 관리 에이전트 시스템과 다른 시스템과의 성능 비교

비교 항목 \ 적용 시스템	집락 시스템 (Zip)	브이박스 시스템 (Vbox)	시멘텍사의 판매 시스템 (Sym)	제안한 시스템 (Ns)
다양한 지불 방법 지원	×	×	×	○
사용자 이익 보호	△	○	△	○
판매자 이익 보호	○	△	○	○
메시지 무결성	×	×	×	○
소프트웨어 불법 복제 방지	×	×	△	○
소프트웨어 불법 유통 방지	×	×	○	○
시스템 확장성	×	×	×	○
네트워크 의존도	△	△	△	○
익명성	○	○	×	×

(○: High △: Low ×: None)

후 관리가 필요 없다. Vbox와 Sym은 제한기간이 지나면 소프트웨어를 시스템에서 삭제 또는 실행 불가로 만들어야 한다. 그러나 Ns의 경우 사용권만을 대상으로 하기 때문에 소프트웨어를 삭제하거나 실행 불가로 만드는 추가적인 노력이 필요 없다.

표 6은 제안한 소프트웨어 지불 관리 에이전트와 다른 소프트웨어 분배 및 구매 시스템과의 성능 평가이다.

Zip, Vbox 그리고 Sym은 사용자가 원하는 다양한 지불 방법을 지원하지 못한다. 그러나 Ns는 다양한 지불 방법을 지원한다. Zip과 Sym은 지불 처리가 먼저 이루어지므로 판매자의 이익이 철저히 보장되는 반면에, 사용자의 이익 보호는 낮다. Vbox는 패키지를 공개 소프트웨어 형식으로 전달하므로 사용자의 이익은 보장되

나, 판매자의 이익 보호가 낮다. Ns에서 사용자는 사용한 시간에 따른 요금을 지불하게 되고, 판매자는 사용자의 불법 복제를 방지 할 수 있으므로 사용자와 판매자의 이익을 동시에 보장한다. Ns는 암호 알고리즘을 이용하여 메시지를 교환하므로 메시지의 무결성이 보장된다. Zip과 Vbox는 잠근 장치를 해제한 소프트웨어에 대하여 무한한 불법 복제 및 유통이 가능하다. Sym은 불법 유통을 막는 효과가 있으나 사용자가 사용권을 다른 사람에게 나누어주는 경우 불법 복제를 막을 수 없다. Ns는 공개키 기반 암호 알고리즘을 이용하여 암호화된 NL을 전달한다. 사용자는 판매자의 공개키(P_{seller})와 자신의 비밀키(S_{user})를 이용하여 평문의 NL을 얻을 수 있고 조작이 가능하다. 그러나 NL을 인증하는 LMA는 암호화된 NL을 복호화하여 사용한다. 따라서 사용자가 NL을 불법 복제하여 배포하더라도, 사용자들은 판매자의 비밀키를 모르기 때문에 불법 복제된 NL로부터 암호화된 NL을 만들 수 없다. 만일 사용자가 암호화된 NL을 제 3자에게 전달하는 경우 자신의 비밀키도 함께 전달해야 한다. 그러나 PKI 구조에서 자신의 비밀키를 노출하는 것은 신용카드의 비밀번호를 노출하는 것과 같이 매우 위험한 일이다. 따라서 Ns는 불법 복제 방지 효과가 크다. Ns는 에이전트로 구현되어 다른 판매자와 사용자와도 연결할 수 있는 확정성을 가진다.

반면에 Ns는 LMA와 UMA간에 SC를 교환하므로 네트워크 의존도가 높고, 불법 복제 및 유통을 방지하기 위하여 사전에 사용자의 정보를 요구하게 되므로 전자상거래의 특징 중의 하나인 익명성을 보장하지 않는다는 단점이 있다.

본 시스템은 기존의 방법보다 인증 및 검사 부분이 추가됨으로서 수행해야 할 알고리즘이 많다. 그러나 알고리즘 복잡도에 있어서 가장 큰 차이가 보이는 것은 역시 암호 알고리즘의 수행이다. 구현한 시스템에서는 RSA 암호 알고리즘을 이용하여 제안한 사용권(NL)과 사용 내역서(SC)를 교환한다. 사용자 측면에서 본다면 소프트웨어가 실행되기 전에 LMA가 암호화된 NL을 복호화 할 때 걸리는 시간을 기다려야 한다. 따라서 실제로 암호화 복호화에 사용되는 시간을 측정해 보았다.

제안한 사용권(NL)과 사용 내역서(SC)는 512 비트의 정형화된 크기를 가진다. 표 7은 512 비트의 데이타를 RSA 알고리즘을 이용하여 본 시스템에서 사용하는 네 가지 경우와 LMA에서 암호화된 NL을 복호화하여 데이타를 얻어내는데 소요된 시간을 10회 측정하여 평균값을 mili-second 단위로 나타낸 것이다. 표 7의 결과를 볼 것 같으면, LMA가 NL을 복호화하여 데이타를 얻어내는데 0.3초가 소요되는 셈이다. 본 시스템이 다른 시스템과 비교하여 소프트웨어 실행시작이 0.3초가 느린 것이다. 그러나 이것은 사용자가 불편을 느낄 만큼 크게 지연되는 시간이 아니므로 본 시스템이 암호 알고리즘을 이용한다고 해서 다른 시스템에 비해 크게 느려진다고 할 수 없다. 그러나 데이타의 크기 또는 키의 길이(X)가 증가하는 경우 연산 수행 시간(N)은 N^X 많큼 증가 하게된다[14,16,17]. 따라서 데이타 NL과 SC의 크기를 늘리지 않는 것이 좋다. 키의 길이 역시 보안의 중요도와 소요되는 시간과의 trade off한 측면을 고려하여 결정하는 것이 좋다.

표 7 구현한 시스템에서 RSA 알고리즘을 수행한 시간

(키의 길이 : 64 bits 단위 : ms)

측정 방법 / 데이터 길이	공개키를 이용한 암호화	비밀키를 이용한 복호화	비밀키를 이용한 전자서명	공개키를 이용한 복호화	LMA가 암호화된 NL을 복호화하여 데이타를 얻는데 소비한 시간
512 bits	53	192	189	46	305

6. 결론 및 향후 연구 과제

기존의 ESD 모델은 소프트웨어 사용에 대하여 다양한 지불 방법을 해결해 주지 못하였다. 본 논문은 이러한 문제를 해결해주는 소프트웨어 사용권 관리 에이전트 시스템을 구현하였다. 제안한 시스템은 NL을 사용하여 기존의 지불 방법 이외에도 사용자가 요구하는 어떠한 지불 방법도 가능하도록 해결하였다. 또한 SC를 제안하여 사용자가 사용한 소프트웨어 값을 판매자가 받는 것을 보장한다. 제안한 시스템은 공개키 암호 방식을 이용하여 NL을 전송하고, 이것을 LMA가 관리하므로 소프트웨어 불법 복제 및 유통을 방지할 수 있다. 제안한 시스템에서 LMA와 UMA는 확장성을 가지고 있어서 판매자 ID를 메시지 다이제스트한 값인 MD5(판매자 ID)로 서로 인증한다. LMA와 UMA간의 이루어지는 SC전달 프로토콜은 atomicity[9,10]한 방법을 사용하여 메시지의 일관성을 보장한다.

제안한 시스템에서는 판매자가 하루에 한번 주기적으로 사용자 정보를 업데이트한다. 업데이트의 주기가 짧을수록 정보의 신뢰도는 높아진다. 그러나 판매자 시스템의 자원을 많이 소모하게 된다. 향후 연구과제로서는 이러한 trade off를 고려한 최적화된 값을 찾아내는 연구가 필요하다.

참 고 문 헌

[1] Kalakota, R. and Whinston, B.A., "Frontiers of Electronic Commerce," IEEE Transactions on Components Packaging & Manufacturing Technology Part C: Manufacturing, Vol.19 No.2, 1996

[2] Yardan, S., "Evaluating the Performances of Electronic Commerce System," Proceedings of the 1997 Winter Simulation Conference, pp.1053-1056, 1997

[3] Jutla, D., Bodorik, P., Hajnal, C., and Davis, C., "Making business sense of electronic commerce," IEEE (us), Computer, Vol.32 No.3, pp.67-75, 1999

[4] ESD, "http://www.previewsystems.com/getstarted/index.html"

[5] ESD models, "http://www.siia.net/pubs/bookstore/items/wpe98.htm"

[6] Masud, S., "Selling bits with Electronic Software Distribution," Intertec Publishing Corporation(us), Vol.23 No.7, 1998

[7] Perlman, R., "An overview of PKI trust models," IEEE Network, Vol.13 No.6, pp.38-43, 1999

[8] Oppliger, R., "Authorization Methods for E-Commerce Applications," Proceedings of the 1999 18th IEEE Symposium on Reliable Distributed Systems, pp.366-371, 1999

[9] Tygar, J.D., "Atomicity in Electronic Commerce," Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, pp.8 - 26, 1996

[10] Billard, D., "Multipurpose Internet Shopping Basket," Proceedings of the Ninth International Workshop on Database and Expert Systems Applications, pp.685-690, 1998

[11] Portland Software, "http://www.portsoft.com/"

[12] ESD, "http://www.newengland-partners.com/RP_DEL5.html#_Toc409344544"

[13] Symantec, "http://www.symantec.com/region/kr/"

[14] Schneier, B., "Applied Cryptography Second Edition," ISBN 0-471-12845-7, pp.28-33, pp.41-44, pp.52-65, pp.429-502, 1996

[15] IAIK, "http://jcewww.iaik.tu-graz.ac.at"

[16] Cramer, R., and Shoup, V., "Signature schemes based on the strong RSA assumption," Proceedings of the 6th ACM conference on Computer and communications security, pp.46 - 51, 1999

[17] Rivest, R.L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," ACM

윤 우 성
1999년 고려대학교 컴퓨터학과 이학사. 1999년 ~ 현재 고려대학교 컴퓨터학과 석사과정. 관심분야는 네트워크, 암호 및 보안, 전자상거래, 멀티미디어 등

윤 정 모
1968년 광운대학교 응용전자공학과 공학사. 1971년 성균관대학교 경영행정대학원 경영학 석사. 1993년 일본 오사카부립대학 공학부 공학박사. 1966년 ~ 1982년 한국전력공사 근무. 1982년 ~ 현재 국립 서울산업대학교 전자계산학과 교수. 국립 서울산업대학원 학과 주임 교수. 관심분야는 시스템 분석 설계, 객체지향 분석 설계, 소프트웨어 공학, 전자상거래, ERP 등

김 태 윤
1981년 고려대학교 산업공학과 공학사. 1983년 미국 Wayne State University 전산과학과 석사. 1987년 미국 Auburn University 전산과학과 박사. 1988년 ~ 현재 고려대학교 컴퓨터학과 교수. 관심분야는 컴퓨터 그래픽스, 네트워크, EDI 시스템, ISDN, 이동통신, 위성통신 등