모바일 환경에서 전자 계약 시스템의 설계 및 구현 : M-XContract

황 기 태† · 김 남 윤††

요 약

모바일 단말기의 하드웨어 한계, 단말기에 탑재된 시스템의 비호환성, 무선의 낮은 대역폭 등으로 인해 유선 환경의 전자 계약 시스템을 M-Commerce에 그대로 적용하기에는 무리가 있다. 본 논문은 이러한 문제점들을 해결하기 위해 XML 기반의 전자 계약서를 정의하고 이를 바탕으로 모바일 환경에 적합한 전자 계약 시스템 M-XContract를 설계 구현한 내용을 다룬다. M-XContract 시스템은 전자 계약 서버, PDA 상에서 고객과 계약하고 전자 서명된 계약서를 서버에 전송하는 M-ESign 모듈, 서명된 전자 계약서를 서버로부터 수신한 후 고객에게 보여주는 M-EDecoder 모듈, 그리고 전자 계약서 저작 도구 X-Auth로 구성된다. M-XContract 시스템의 성능을 분석하기 위해 PDA 상에서 전자 서명 생성 시간, 서버로 전송 시간을 측정하였다. 성능 분석 결과 M-XContract 시스템은 모바일 전자 계약 시스템을 위한 효과적인 모델이라는 결론을 얻었다.

Design and Implementation of Digital Contract System on the Mobile Environment : M-XContract

Kitae Hwang† · Namyun Kim††

ABSTRACT

Due to hardware resource limit and system incompatibility of the mobile device, and low bandwidth of wireless communication, there are a few difficulties in introducing the digital contract system based on wired communication to M-Commerce. To get over the difficulties, this paper defines a digital contract based upon XML and then addresses the design and implementation of M-XContract, a digital contract system for the mobile environment. M-XContract system has been constructed with the digital contract server, M-ESign module which contracts with the customer on the PDA and transfers the contract digitally signed to the server, M-EDecoder module which shows the contract to the customer from the server, and X-Auth which is a contract authoring tool. To evaluate the run-time performance of the M-XContract, we measured the digital signature generation time and transfer time to the server. The evaluation results show that the M-XContract is an efficient model for the mobile contract system.

키워드 : 전자 서명(Digital Signature), 모바일(Mobile), XML, PKI, 전자 계약서(Digital Contract), M-Commerce

1. 서 론

지난 몇 년간 하드웨어의 소형화 기술 및 네트워크 기술의 획기적 발전으로 인해 Cellular Phone, PDA(Personal Digital Assistant), Smart Phone과 같은 모바일 단말기들은 고성능화되고 사용이 증가되어 왔다. 이에 따라 모바일 전자 계약(M-Contract), 모바일 뱅킹(M-Banking), 모바일 금융 거래(M-Payment) 등 다양한 종류의 M-Commerce가 시작되고 있다[1, 2]. 아직은 초기 단계이지만 모바일 단말기의 편리성, 신속성, 시공간을 초월할 수 있는 장점들로 인해 M-Commerce 분야는 점차 활성화될 것으로 전망된다.

본 논문은 상기 M-Commerce의 응용 중에서 모바일 전자 계약을 연구의 대상으로 설정하였다. 모바일 전자 계약이란 영업사원이 모바일 단말기를 이용하여 상품 계약서를 작성하거나, 개인이 자동차 보험 가입이나 카드 발급 신청을 수행하는 행위를 통칭한다. 모바일 전자 계약은 자동차 보험, 생명 보험과 같은 방문 보험 계약, 현지에서의 농산물 직거래 계약, 방문 카드 가입 등 사회 각 분야에서 매우 넓은 응용 분야를 차지한다.

그러나 모바일 전자 계약을 비롯한 M-Commerce는 PC를 기반으로 하는 기존의 인터넷 전자 상거래와는 달리 시스템의 설계시 고려되어야 하는 세 가지 문제점이 존재한다. 첫째, 모바일 단말기는 제한된 CPU 처리 능력 및 메모리를 가진다. 예를 들어 Cellular Phone은 33MHz CPU와 4MB RAM, PDA는 200MHz CPU와 64MB RAM 정도를 가지고 있는 것이 일반적이다. 따라서 M-Commerce에서 요구되는 암호화, 전자 서명, 사용자 인증 등의 보안 처리에 소요되는 시간이 매우 크다는 단점이 존재한다. 둘째, 유선 통신과 비교하여 모바일 통신은 대역폭이 매우 낮으며 높은 에러율을 가진다. 현재 상용화되고 있는 CDMA 2000 1x는 최대 144kbps이며

※ 본 연구는 2003학년도 한성대학교 공학연구센터 특별 연구비 지원 과제임.
† 정 회 원 : 한성대학교 컴퓨터공학부 교수
†† 정 회 원 : 한성대학교 정보공학부 교수
논문접수 : 2003년 5월 19일, 심사완료 : 2003년 8월 25일

GPRS 통신망은 최대 171kbps의 속도를 가진다. 그러나 실제 전송 속도는 최대 속도보다 낮으며 GPRS 평균 속도는 20~30kbps 정도인 것으로 알려져 있다[3]. 이러한 송수신 속도의 지연 및 불예측성은 실시간 처리를 어렵게 한다. 셋째, 모바일 단말기는 Cellular Phone, PDA, Handheld PC와 같이 다양한 종류가 존재하며 여기에 탑재되는 시스템 소프트웨어나 응용 분야 또한 다양하다. 따라서 단말기 사이에 호환성을 유지할 수 있는 M-Commerce 시스템 개발이 필요하다. 결국, 유선에서 사용되는 전자 계약 시스템을 모바일 환경에 그대로 적용하기에는 여러 가지 문제가 존재한다.

본 논문은 이러한 문제점으로 해결하기 위해 특정 파일 포맷에 의존하지 않으며 크기가 작은 XML 기반 전자 계약서를 정의한다. 전자 계약 시스템은 본질적으로 전자 서명을 바탕으로 하며 본 논문에서는 이를 위해 PKI 기반 전자 서명 모델을 이용한다. 이들을 바탕으로 모바일 전자 계약 시스템 모델인 M-XContract과 모바일 전자 계약시 사용되는 보안 모델을 정의하고 이를 설계 구현한 상세 내용을 보인다. 마지막으로 전자 서명 및 모바일 환경에서 계약서의 전송에 따른 M-XContract의 실행 성능을 평가한다.

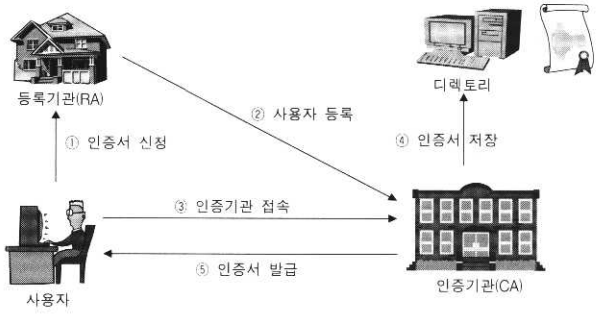
본 논문의 구성은 다음과 같다. 2장에서 본 연구의 배경이 되는 PKI 기반 전자 서명 모델 및 관련 연구에 대해 설명하고 3장에서는 XML 기반 전자 계약서를 정의한다. 그리고 4장에서는 본 논문에서 구현한 전자 계약 시스템인 M-XContract에 대해 상세히 논하며 5장에서는 M-XContract의 성능을 측정 평가한다. 마지막으로 6장에서 결론을 맺는다.

2. 연구 배경

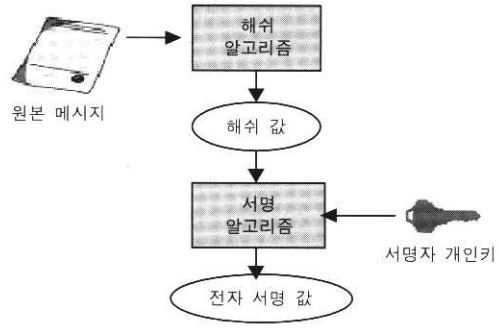
2.1 PKI 기반 전자 서명

암호 시스템은 일반적으로 대칭형 암호 시스템(symmetric cryptosystem)과 공개 키 암호 시스템(public-key cryptosystem)으로 나뉘어진다[4]. 대칭형 암호 시스템은 암호화 및 복호화시 동일한 키를 사용하며 공개 키 암호 시스템은 암호화와 복호화시 공개 키(public key)와 개인 키(private key)를 사용하는 비대칭형 암호 시스템이다. 따라서 공개 키 암호 시스템은 사용자가 공개 키와 개인 키 쌍을 보유하고 있어야 하며, 공개 키는 모든 사용자가 접근 가능한 공개된 장소에 등록하고 개인 키는 안전한 장소에 보관되어야 한다.

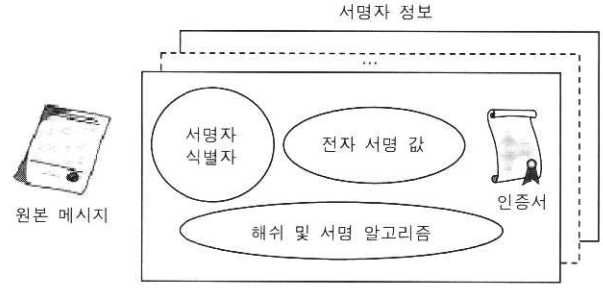
한편 공개 키는 임의의 사용자가 수정하는 것을 방지하기 위해 인증서(certificate)에 저장된다. 인증서는 사용자의 신원과 공개 키를 연결해주는 문서로서 인증 기관(Certificate Authority)이 자신의 개인 키로 서명하여 발급된다. 일반적으로 인증서는 ITU(Internet Telecommunication Union)에서 제시한 X.509의 형식을 따르고 있으며 인증서의 발급, 분배, 관리를 위한 시스템을 공개 키 기반 구조(Public Key Infrastructure)라고 한다[5]. 즉, PKI는 공개 키 암호 시스템을 이용하기 위한 기반 구조를 제공해주는 솔루션이며 PKI를 구성하고 있는 개체로는 (그림 1)과 같이 인증 기관, 등록 기관, 인증서를 저장하고 있는 디렉토리 등으로 구성된다.



(그림 1) PKI 구성 요소



(a) 전자 서명 생성 과정



(b) 다수의 사용자가 서명 하였을 때의 서명 데이터 내용

(그림 2) PKI 기반 전자 서명

본 연구는 PKI 기반 전자 서명 시스템을 대상으로 한다. 전자 서명(Digital Signature)은 계약서나 약정서에 서명을 생성함으로써 계약 내용을 위조하거나 부인을 봉쇄할 수 있는 기능을 제공하며 주요 서명 알고리즘으로는 RSA, DSA, ECDSA 등이 존재한다[4]. 일반적으로 많이 사용되고 있는 RSA 전자 서명을 생성하는 과정은 (그림 2)와 같이 크게 두 단계로 나눌 수 있다. 첫 번째 단계에서는 해쉬 알고리즘을 사용하여 원본 메시지를 압축한다. 해쉬 알고리즘은 임의 크기의 메시지를 고정된 비트의 값으로 압축하는 것으로서 MD5[6], SHA-1[7] 등이 있다. 두 번째 단계에서는 압축된 해쉬 값을 서명자의 개인 키로 암호화하여 전자 서명을 생성한다. (그림 2)(b)는 하나의 원본 메시지에 대해 다수의 사용자가 서명을 하였을 경우 서명 데이터 내용을 보여주고 있다. 서명 데이터(Signed Data)[8]에는 원본 메시지와 서명자 정보들로 구성된다. 그리고 서명자 정보에는 서명자 식별자, 해쉬 및 서명 알고리즘 식별자, 전자 서명

값, 서명자의 인증서 등이 포함된다.

한편 전자 서명을 검증하는 작업은 두 단계로 나눌 수 있다. 첫 번째 단계에서는 서명 데이타 내의 원본 메시지를 압축하여 해쉬 값을 계산한다. 두 번째 단계에서는 역시 서명 데이타 내의 전자 서명 값을 서명자의 인증서에 저장된 공개키를 사용하여 복호화하여 해쉬 값을 얻는다. 그리고 이 두 개의 해쉬 값을 비교하여 서명이 올바른지 검증한다. 한편, 서명자의 인증서가 유효 기간 전에 폐기 되었는지의 유무를 파악하기 위해 CRL(Certificate Revocation List)[9]이나 OCSP (Online Certificate Status Protocol)[10] 등이 사용된다.

2.2 관련 연구

지금까지 M-Commerce에 관한 연구는 모바일 환경의 제약 조건을 해결하는데 초점을 맞추어 진행되어 왔다[10-13]. 첫째, 모바일 단말기의 제한된 메모리의 특성을 고려하여 인증서의 형식을 간소화하였다. 즉, 유선 환경에서는 X.509 인증서를 사용하였으나 모바일 환경에서는 항목의 수가 작고 키 값과 서명의 크기를 줄일 수 있는 WTLS 인증서가 제안되었다[12]. 둘째, 모바일 단말기의 낮은 대역폭을 고려하여 효율적인 인증서 검증 기법이 제안되었다. 즉, 유선 환경에서는 인증서의 취소 유무를 검증하기 위해 인증서 취소 목록 CRL을 사용하는데, 이는 모바일 환경에서 많은 전송 시간을 유발한다. 따라서 모바일 환경에서는 "Short Lived Certificate"[12]을 사용하거나 OCSP 등의 방법이 제안되었다. 셋째, 모바일 단말기의 제한된 CPU 성능을 고려하여 효율적인 서명 알고리즘을 개발하였다. 현재 유선 환경에서는 1024비트 키를 가지는 RSA 알고리즘이 많이 사용되고 있는데, 이는 휴대폰과 같은 낮은 CPU 성능을 가지는 환경에서는 서명 생성시 수 초 이상의 많은 시간이 소요된다. 따라서 모바일 환경에서는 RSA 알고리즘과 동일한 보안을 만족시킬 수 있으면서 효율적인 타원 곡선 알고리즘이 개발되었다[13].

최근 전자 서명 문서의 포맷, 사용자 인터페이스, 전체 시스템의 효율적인 구성 등에 관한 연구가 진행되고 있다. 다양한 운영체제(WinCE, Palm, Symbian, Proprietary OS)가 사용되는 모바일 환경에서는 전자 문서의 호환성을 높이고 전자 문서의 크기를 줄일 수 있는 연구가 필수적이라고 볼 수 있다. 따라서 본 논문은 모바일 환경에서 호환성이 뛰어나고 크기가 작은 XML 기반 전자 계약서를 정의하고 효율적인 전자 계약 시스템의 구현 모델을 제시하고자 한다.

3 전자 계약서

3.1 XML 문서

일반적으로 계약서는 워드나 아래 한글과 같은 범용 문서 편집기를 이용하여 작성된다. 그런데 이러한 계약서를 모바일 전자 계약 시스템에서 사용하기에는 몇 가지 문제점이 존재한다. 첫째, 실제 계약서 내용을 포함하는 텍스트 길이에 비해 파일의 크기가 매우 크다. 예를 들어 마이크로소프트사의 워드 7.0의 경우 단지 한 글자만 입력하고 저장해도 19KB나 된다. 그러므로 이러한 범용 문서 편집기로 저작된 계약서를 사용하면 모바일 단말기의 메모리를 낭비하고 서버와의 계약서 송수신 시간이 길어지므로 응답 시간이 증가하는 단점이 있다. 둘째, 범용 문서 편집기로 작성된 계약서 문서는 단말기 종류에 따라 호환되지 않는 문제점을 가지고 있다. 셋째, 모바일 전자 계약서에는 많은 경우 다양한 사용자 인터페이스가 요구된다. 즉, 가입자에 대한 정보를 입력하거나 계약 옵션의 선택 등 사용자 인터페이스가 필요하다. 그러나 텍스트 중심의 범용 문서 편집기는 콤보 박스나 라디오 버튼과 같이 다양한 인터페이스를 지원하지 못하는 단점이 존재한다.

본 논문에서는 이러한 단점을 극복하기 위해 XML 기반 전자 계약서를 정의한다. XML(eXtensible Markup Language)은 구조화된 문서와 데이터를 표현하기 위한 마크업 언어이다[14]. 현재 많은 응용 프로그램들이 자신의 데이터 구조를 표현하기 위해 XML을 사용하고 있다. MathML(Mathmatical Markup Language)은 두 컴퓨터 사이의 전송할 수학적 표현을 위해 설계된 경우이며, XForms는 XML을 이용하여 HTML 문서의 폼을 개선한 경우이다. 전자 계약서를 위해 필요한 폼 구성 요소는 그리 복잡하지 않기 때문에, 전자 계약서의 형식을 위해 XForms의 정의를 사용하기에는 복잡하고 높은 부하의 부담을 가진다. 현재 모바일 전자 계약서를 위해 제안된 XML 표준이 없기 때문에 본 논문에서는 전자 계약서를 위해 새로운 XML 태그 셋을 정의하고 전자 계약서 저작 도구(X-Auth)를 개발 구현하였다. 본 논문에서 정의된 XML 기반 전자 계약서는 윈도우 컨트롤을 이용한 다양한 인터페이스를 지원한다.

〈표 1〉 M-XContract 전자 계약서의 주요 XML 태그

요소(element)	기 능	자식(child) 요소
<contract>	계약서 문서의 XML 폼의 루트	<header>, <form>
<header>	헤더 정보	<title>, <id>
<form>	계약서의 모양	<area>+
<titile>	계약서 제목	없음
<id>	계약서 id	없음
<area>	문서 내의 한 영역을 정의. 문서는 여러 개의 영역으로 분할. 영역의 기준은 모바일 단말기의 화면 크기를 기준으로 설정.	<textlabel>*, <input>*, <datelabel>*, <signlabel>*, <linelabel>*, <rectanglelabel>*, <circlelabel>*
<textlabel>	텍스트 스트링	없음
<input>	사용자 입력(텍스트, 콤보 박스, 라디오 버튼 등)	<option>*
<option>	콤보 박스나 라디오 버튼의 선택 항목	없음
<datelabel>	계약에 관련된 날짜 정보	없음
<signlabel>	전자서명 버튼	없음
<linelabel>	선	없음
<rectanglelabel>	사각형	없음
<circlelabel>	원	없음

+ : 반복 가능하며 반드시 필요한 요소
* : 반복 가능하며 선택적으로 필요한 요소

3.2 M-XContract 전자 계약서

모바일 전자 계약서를 위한 XML 태그 셋을 정의함에 있어 다음 두 가지가 고려되었다.

1) 모바일 전자 계약서의 구성 요소 : 인터넷이나 오프라인에서 사용되는 보험 계약서, 카드 가입 약정서 등을 분석한 결과, 모바일 전자 계약서를 구성하는 요소를 ① 계약서의 본문을 위한 *텍스트 스트링*, ② 계약서를 작성하는 고객이 입력할 *입력 상자들*, ③ 계약일 등의 *날짜 정보*, ④ 선, 원, 사각형 등의 *도형*, 그리고 ⑤ *서명 버튼*으로 정의하였다. 입력 상자는 콤보 박스, 라디오 버튼, 체크박스 3가지를 정의하였다. 이들을 표현하기 위해 정의된 주요 XML 태그들은 <표 1>과 같다. <id> 요소는 고객 DB 속에 입력된 사용자 ID를 입력하는 목적으로 정의되었다. M-XContract 시스템에서 서명된 전자 계약서는 (그림 2)(b)의 모델을 따르며 그림에서 원본 메시지가 바로 XML 전자 계약서의 텍스트 스트링이다.

2) 모바일 전자 계약서의 화면 출력 : 현재 본 논문의 모바일 단말기로 사용된 Compaq iPAQ의 경우 액정 화면이 320×240이고 다른 PDA의 경우도 이와 유사한 크기로 작은 편이다. 따라서 단말기 화면의 크기를 고려하지 않고 계약서가 작성된다면 고객에게 불편함을 주게 된다. 본 논문에서는 이러한 문제를 해결하기 위해 전자 계약서의 XML 요소로 <area> 태그를 정의하였다. <area> 요소는 계약서 내용 중 모바일 단말기 화면 내에 한 번에 보여줄 수 있는 크기로 한 페이지 혹은 계약서 상의 의미에 따라 나눌 수 있는 영역을 정의한다.

(그림 3)은 본 논문에서 정의된 XML 태그 셋을 이용하

```
<?xml version = "1.0" encoding = "EUC-KR"?>
- <contract>
  - <header e-sign = "no">
      <title text = "자동차 보험 계약서"/>
      <id name = " "/>
    </header>
  - <form stripes = "4">
      <labelNumber textlabel = "16" styledtextlabel = "1" signlabel = "2" linelabel = "1" circlelabel = "1" rectlabel = "1" roundrectlabel = "1"
        inputlabel = "14" radiolabel = "6">
    - <area name = "customer" width = "200" height = "151" labelswidth = "180" labelsheight = "131">
        <textlabel name = "textlabel 1" x = "10" y = "60" width = "60" height = "21" fontsize = "12" text = "고객성명"/>
        <input type = "box" fontsize = "12" text = " " name = "input 1" x = "80" y = "60" width = "110" height = "21"/>
        <textlabel name = "textlabel 2" x = "10" y = "90" width = "60" height = "21" fontsize = "12" text = "전화번호"/>
        <input type = "box" fontsize = "12" text = " " name = "input 2" x = "80" y = "90" width = "110" height = "21"/>
        <textlabel name = "textlabel 3" x = "10" y = "120" width = "60" height = "21" fontsize = "12" text = "주소"/>
        <input type = "box" fontsize = "12" text = " " name = "input 3" x = "80" y = "120" width = "110" height = "21"/>
        <textlabel name = "textlabel 15" x = "30" y = "10" width = "141" height = "30" fontsize = "15" text = "자동차 매매 계약서"/>
      </area>
    - <area name = "car" width = "200" height = "188" labelswidth = "180" labelsheight = "171">
        <textlabel name = "textlabel 4" x = "10" y = "10" width = "60" height = "21" fontsize = "12" text = "차종"/>
      + <input type = "combo" selecteditem = "4" fontsize = "12" text = " " name = "input 4" x = "80" y = "10" width = "110" height = "21"/>
        <textlabel name = "textlabel 5" x = "10" y = "40" width = "60" height = "21" fontsize = "12" text = "색상"/>
      + <input type = "combo" selecteditem = "5" fontsize = "12" text = " " name = "input 5" x = "80" y = "40" width = "110" height = "21"/>
        <textlabel name = "textlabel 6" x = "10" y = "70" width = "60" height = "21" fontsize = "12" text = "변속기"/>
      - <input type = "radio" selecteditem = "1" name = "input 6">
          <option name = "radio 1" x = "80" y = "70" width = "50" height = "21" fontsize = "12" text = "수동" />
          <option name = "radio 2" x = "140" y = "70" width = "50" height = "21" fontsize = "12" text = "오토" />
        </input>
        <textlabel name = "textlabel 7" x = "10" y = "100" width = "60" height = "21" fontsize = "12" text = "선택사항"/>
        <input type = "multilinebox" fontsize = "12" text = " " name = "input 7" x = "80" y = "100" width = "110" height = "81"/>
      </area>
    - <area name = "payment" width = "200" height = "157" labelswidth = "180" labelsheight = "141">
        <textlabel name = "textlabel 8" x = "10" y = "10" width = "60" height = "21" fontsize = "12" text = "차량가격"/>
        <input type = "box" fontsize = "12" text = " " name = "input 8" x = "80" y = "10" width = "110" height = "21"/>
        <textlabel name = "textlabel 9" x = "10" y = "40" width = "60" height = "21" fontsize = "12" text = "선택가격"/>
        <input type = "box" fontsize = "12" text = " " name = "input 9" x = "80" y = "40" width = "110" height = "21"/>
        <textlabel name = "textlabel 10" x = "10" y = "70" width = "60" height = "21" fontsize = "12" text = "합계"/>
        <input type = "box" fontsize = "12" text = " " name = "input 10" x = "80" y = "70" width = "110" height = "21"/>
        <textlabel name = "textlabel 11" x = "10" y = "100" width = "60" height = "21" fontsize = "12" text = "지불방법"/>
      + <input type = "radio" selecteditem = "1" name = "input 11">
        <textlabel name = "textlabel12" x = "10" y = "130" width = "60" height = "21" fontsize = "12" text = "할부횟수"/>
      + <input type = "combo" selecteditem = "5" fontsize = "12" text = " " naem = "input 12" x = "80" y = "130" width = "110" height = "21"/>
      </area>
    - <area name = "sign" width = "200" height = "107" labelswidth = "180" labelsheight = "91"/>
        <textlabel name = "textlabel 13" x = "10" y = "10" width = "60" height = "21" fontsize = "12" text = "인수일"/>
        <input type = "box" fontsize = "12" text = " " name = "input 13" x = "80" y = "10" width = "110" height = "21"/>
        <textlabel name = "textlabel 14" x = "10" y = "40" width = "60" height = "21" fontsize = "12" text = "계약일"/>
        <datelabel type = "sign" fontsize = "12" text = " " name = "datelabel 1" x = "80" y = "40" width = "110" height = "21"/>
        <signlabel name = "signlabel 1" x = "70" y = "70" width = "60" height = "31" fontsize = "12" text = "서명"/>
      </area>
    </form>
</contract>
```

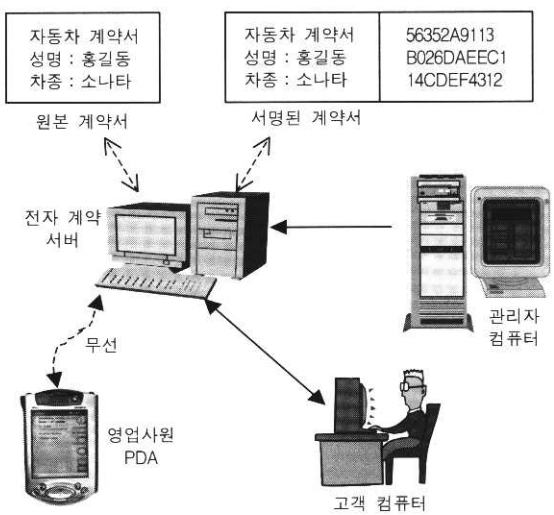
(그림 3) 자동차 매매 계약서의 XML

여 작성된 자동차 매매 계약서의 예를 보여주며 (그림 13)은 X-Auth를 이용하여 자동차 매매 계약서를 실제 작성하는 예를 보여준다.

4. M-XContract 시스템

4.1 시스템 모델

본 논문에서 구현한 모바일 전자 계약 시스템 M-XContract는 (그림 4)와 같이 모바일 단말기, 고객 컴퓨터, 전자 계약 서버, 관리자 컴퓨터로 구성되며 사양은 <표 2>와 같다.



(그림 4) M-XContract 시스템 모델

〈표 2〉 M-XContract 시스템의 사양

시스템 요소		물리적 요소
PDA	모델	Compaq iPAQ 3850, 3950
	무선 인터넷	Mobile Explorer
	운영체제	PocketPC 2002
전자 계약서버	하드웨어	펜티엄급 PC
	운영체제	윈도우 2000 서버
	웹 서버	IIS
	JSP 엔진	Tomcat
	DBMS	SQL Server 7.0
	전자 계약서	XML 폼(본 논문에서 정의)
고객 컴퓨터	모델	웹 브라우저를 갖춘 PC
관리자 컴퓨터	모델	웹 브라우저를 갖춘 PC
	전자 계약서 저작 도구	본 논문에서 구현된 전자 계약서 저작 도구(X-Auth)

전자 계약 서버는 원본 계약서와 고객의 정보 및 전자 서명이 담긴 서명된 계약서를 관리한다. 이들 계약서는 데이터베이스의 서로 다른 테이블에 별도 저장되어 관리된다. 그리고 PDA, 고객 컴퓨터, 관리자 컴퓨터 등과 웹을 이용하여 통신하는 JSP(Java Sever Page) 페이지들을 가지고 있다. 서명된 계약서는 고객별로 관리되며 고객은 웹에서 자신의 계정을 이용하여 서버로부터 언제든지 확인할 수 있다. 다수의 PDA가 무선으로 동시에 전자 계약 서버에 접속 가능하다. 관리자는 본 연구에서 개발 구현한 X-Auth라는 저작 도구를 사용하여 원본 전자 계약서를 편집하고 이를 전자 계약 서버에 전송하는 역할을 한다.

M-XContract 시스템에서 PDA 소유자는 고객이 아닌 영업 사원이므로 고객은 플로피 디스크나 USB 토큰 등의 매체를 통해 개인 키와 인증서를 가지고 있어야 한다. 즉, 상용화를 위해서는 고객이 자신의 인증서와 개인 키를 저장한 별도의 저장 장치를 가지고 있어야 한다. 그러나 M-XContract 시스템은 저장 장치의 위치와 큰 관련이 없다. 즉, 전자 서명 생성시 개인 키 및 인증서가 필요하기 때문에 서명 모듈의 일부 파라미터를 수정하면 된다. 따라서 본 시스템의 초기 프로토타입에서는 고객의 개인 키와 인증서가 모바일 단말기의 메모리에 존재하는 것을 가정하였다.

4.2 동작 모델

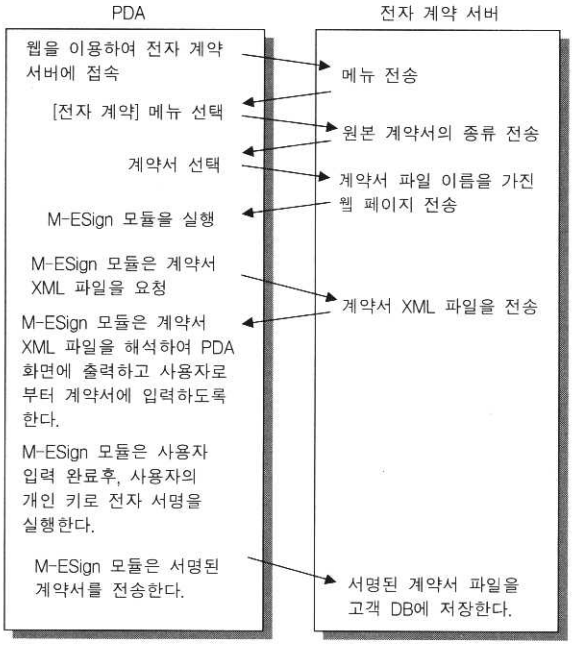
4.2.1 소프트웨어 구성

M-XContract 시스템을 구성하는 소프트웨어 모듈은 크게 4 부분이다 : *서버 상의 웹 페이지들, 전자 계약서 저작 모듈, M-ESign 모듈, M-EDecoder 모듈.*

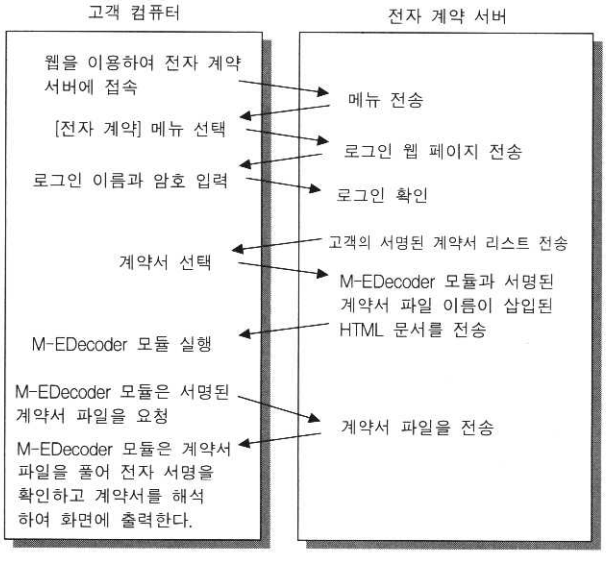
모바일 단말기의 사용자나 고객이 전자 계약 서버에 접근하기 위한 웹 페이지들은 서버에서 JSP로 구현되며 전자 계약서 저작 모듈인 X-Auth는 XML 형태로 전자 계약서를 만들 수 있는 도구이다. M-ESign은 모바일 단말기의 ME (Mobile Explorer) 브라우저에 내장되어 수행되는 ActiveX 컨트롤이다. M-ESign 모듈은 전자 계약 서버로부터 XML 문서를 전송 받은 후 PDA 화면에 출력하고 사용자 입력을 받고, 계약서에 전자 서명하여 서명된 계약서를 전자 계약 서버에 전송하는 역할을 수행한다. 고객은 자신의 계약을 확인하기 위하여 고객 컴퓨터를 이용하여 전자 계약 서버에 접속하면, 익스플로러에 의해 웹 페이지에 내장된 M-EDecoder 모듈이 서버로부터 다운로드되고 실행된다. M-EDecoder 모듈의 기능은 전자 계약 서버로부터 서명된 계약서를 전송 받고 전자 서명을 확인하고 계약서를 웹 브라우저 화면에 출력하는 것이다. M-ESign 모듈은 PDA 내에 기본적으로 설치되어 있어야 하지만 M-EDecoder 모듈은 전자 계약 서버로부터 웹 페이지와 함께 다운로드되어 고객의 컴퓨터에 설치된다. 고객이 전자 계약 서버에 접속할 때마다 고객의 컴퓨터에 가장 최근에 설치된 M-EDecoder의 버전과 비교하여 버전이 갱신되었으면 다시 전자 계약 서버로부터 M-EDecoder가 다운로드된다.

4.2.2 동작

전자 계약을 위해 PDA와 전자 계약 서버 사이의 동작은 (그림 5)와 같고, 고객 컴퓨터와 전자 계약 서버 사이의 동작 과정은 (그림 6)과 같이 모델링된다.



(그림 5) 전자 계약 서버와 PDA 사이의 통신



(그림 6) 전자 계약 서버와 고객 컴퓨터 사이의 통신

4.3 M-ESign 모듈

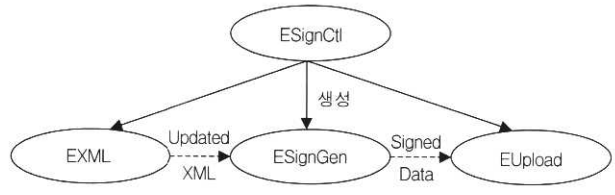
PDA에서 수행되는 M-ESign 모듈은 경량의 ATL(Active Template Library)[15]을 사용하여 ActiveX 컨트롤로 구현되었다. M-ESign 모듈은 DLL 형태로 구현되었으며 ME와 동일한 프로세스 영역에서 수행되는 In-Process Server로 구현되었다.

(그림 7)은 PDA의 ME가 서버로부터 전송 받은 HTML 문서를 보여주고 있다. ME는 classid 속성에 지정된 clsid 값을 가지는 M-ESign 모듈을 시스템 레지스트리(Registry)로부터 찾아 ME의 화면 공간 내에 실행시킨다. 그리고 <param> 태그의 name 속성의 값(원본 계약서 파일의 URL)을 M-ESign 모듈에 전달하게 된다. M-ESign 모듈은 이 원본 XML 전자 계약서를 서버로부터 다운로드 받아 출력하고 사용자 입력을 받은 후 전자 서명하여 서버로 전송한다.

```
< object classid = "clsid : M-ESign의 clsid" width = "250"
        height = "500">
< param name = "FileName" value = "원본 계약서 파일의 URL" >
< /object >
```

(그림 7) PDA ME가 서버로부터 전송받는 HTML 문서 일부

M-Esign 모듈의 주요 클래스는 다음과 같다(그림 8).



(그림 8) M-ESign 모듈의 구성 요소

① M-ESign 클래스(ESignCtl)

HTML 문서의 <Object> 태그에 명시된 ActiveX 컨트롤로서 PDA의 ME에 의해 로드된다. 이 컨트롤은 다른 클래스를 생성하는 역할을 수행한다.

② XML 처리 클래스(EXML)

서버로부터 XML 전자 계약서를 수신한 후 XML DOM (Document Object Model) API(MSXML Lib)를 이용하여 XML 문서의 노드를 추출한다. 그리고 윈도우 컨트롤(Combo Box, Radio Button)을 이용하여 XML 문서 노드를 PDA 화면에 디스플레이한다. 그리고 사용자가 입력한 내용을 받아 XML 문서를 수정한다.

③ 전자 서명 클래스(ESignGen)

서명자의 개인 키를 이용하여 XML 문서에 대한 전자 서명 값을 생성하는 부분이다. 즉, WinCE 기반의 전자 서명 라이브러리(CryptoLib)를 이용하여 XML 문서를 해쉬 및 암호화하여 전자 서명한다. 그리고 서명자의 인증서, 전자 서명, XML 문서 등을 DER(Distinguished Encoding Rules)을 이용하여 인코딩하여 서명 데이터를 생성한다.

④ 파일 전송 클래스(EUpload)

서명 데이터를 서버로 업로드하는 클래스로서 WinInet 라이브러리를 이용한다. WinInet은 마이크로소프트에서 제공하는 라이브러리로서 HTTP, FTP, Gopher 프로토콜을 지원한다. 이 요소는 WinInet의 HTTP 프로토콜을 이용하여 서버에 서명 데이터를 전송한다.

4.4 M-EDecoder 모듈

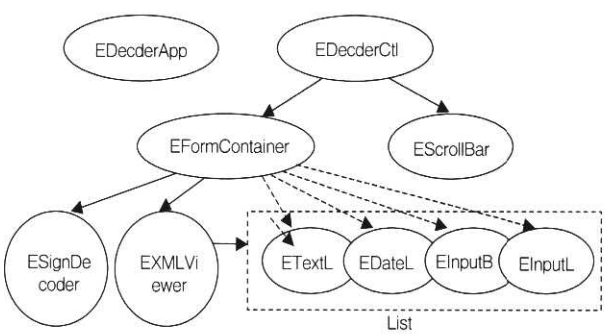
M-EDecoder 모듈은 고객 컴퓨터에서 수행되며 고객이 자

신의 서명한 계약서를 확인하기 위해 존재한다. 이 모듈 역시 마이크로소프트 사의 COM 모듈로서 MFC 라이브러리를 이용하여 ActiveX 형태로 작성되었다. 웹 브라우저는 전자 계약 서버로부터 수신한 (그림 9)와 같은 HTML 문서에 지정된 M-EDecoder 모듈을 실행한다. M-EDecoder 모듈은 <param> 태그에 주어진 서명된 계약서 파일의 URL을 얻고 계약서 파일을 전자 계약 서버로부터 다운로드 받는다. 그리고 전자 서명을 확인한 후 윈도우 컨트롤을 이용하여 고객 컴퓨터 화면에 출력한다.

```
< object classid = "clsid : M-EDecoder의 clsid"
        codebase = "M-EDecoder.cab">
< param name = "FileName" value = "서명된 계약서 파일의 URL" >
</object>
```

(그림 9) 고객 컴퓨터가 서버로부터 전송받는 HTML 문서 일부

(그림 10)은 M-EDecoder 모듈의 실행 시간 객체 구성도를 보여준다.



(그림 10) M-EDecoder 모듈의 실행 시간의 객체 구성도

(그림 10)의 각 객체들은 각각 <표 3>의 클래스 인스턴스들이다. EDecoderCtl 객체 인스턴스는 ActiveX 컨트롤이며 자신이 생성될 때 EFormContainer 인스턴스와 EScrollBar 인스턴스를 자식으로 생성한다. EFormContainer 인스턴스는 계약서를 출력할 컨테이너 역할을 수행하며 EScrollBar 인스턴스에 의해 스크롤 제어된다.

EFormContainer 인스턴스는 ESignDecoder 객체 인스턴스를 생성하여 서명된 계약서로부터 XML 계약서를 분리하게 하고, 이 XML 스트링을 EXMLViewer 인스턴스에 넘겨주어 계약서를 화면에 출력하게 한다. <표 1>의 각 XML 태그들은 (그림 10)에서 점선으로 둘러싸인 박스 내의 인스턴스와 각각 대응되며 <표 3>의 ELabel을 상속받는 각 클래스들과 대응된다.

4.5 전자 계약 서버

전자 계약 서버는 기본적으로 세 개의 DB 테이블을 유지 관리한다. 이들은 고객의 이름과 암호를 관리하는 고객 DB 테이블, 원본 계약서 DB 테이블, 서명된 계약서 DB 테이블이며, 계약서 DB 테이블은 (그림 11)과 같이 설계되었다. 원본 계약서와 서명된 계약서를 파일로 각각 유지하며 DB 테이블은 이들의 파일 이름만을 가지고 있도록 설계되었다.

계약서 이름	계약서 작성일	XML 계약서 파일 이름

(a) 원본 계약서 DB 테이블

고유번호(키)	계약서 이름	계약자 로그인 ID	날짜	서명된 계약서 파일 이름

(b) 서명된 계약서 DB 테이블

(그림 11) 원본 계약서와 서명된 계약서 DB 테이블

〈표 3〉 M-EDecoder를 구성하는 주요 클래스

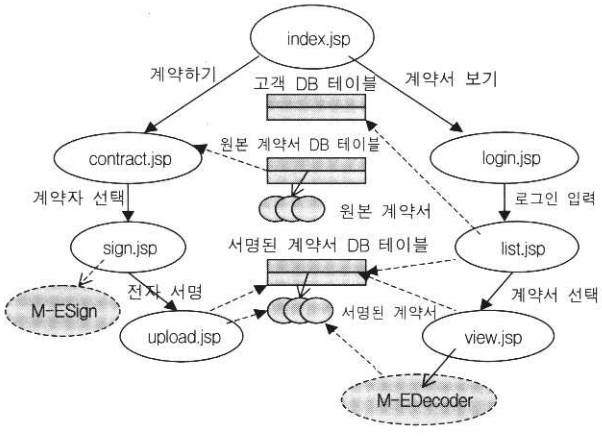
클 래 스	기반 클래스	기 능
EDecoderCtl	Static 컨트롤을 서브클래싱	ActiveX 컨트롤 인터페이스를 구현하며 ActiveX를 포함하는 컨테이너와 통신하는 인터페이스를 구현한다.
EDecoderApp		COM 모듈을 레지스트리에 등록, 해제한다.
EFormContainer	CStatic	계약서가 출력되는 공간의 컨테이너 역할을 하는 static 컨트롤 클래스
EScrollBar	CScrollBar	스크롤바 기능으로 EFormContainer를 스크롤
ESignDecoder		WinInet을 이용하여 HTTP 연결을 도모하고 서명된 계약서를 서버로부터 다운로드 받은 후 서명을 확인하고 계약서 XML을 분리한다.
EXMLViewer		계약서 XML 스트링을 분석하여 계약서를 화면에 출력한다.
ELabel	CStatic	전자 계약서를 구성하는 레이블 객체들의 조상 클래스
ETextLabel	ELabel	단순 텍스트 스트링을 표현하는 클래스
EInputLabel	ELabel	계약서에서 사용자로부터 입력 받는 컨트롤
EInputBoxLabel	EInputLabel	계약서 상의 텍스트 입력 창을 표현하는 컨트롤
EInputComboLabel	EInputLabel	계약서 상의 콤보 박스를 표현하는 컨트롤
EInputRadioLabel	EInputLabel	계약서 상의 라디오 버튼을 표현하는 컨트롤
EDateLabel	ELabel	계약서 상의 날짜 정보를 표현하는 컨트롤

전자 계약 서버의 주어진 기능은 서버가 관리하는 계약서들에 접근하는 과정을 지원한다. 서버가 지원하는 각 JSP 페이지들의 상태 천이도는 (그림 12)와 같다.

JSP 페이지들은 서버에서 실행되는 코드로서 웹 클라이언트에(PDA ME, 고객 컴퓨터의 브라우저) HTML 소스를 출력하는 기능을 수행한다.

"index.jsp"는 웹 클라이언트에 [전자 계약]와 [계약서 보기] 메뉴를 가진 HTML 코드를 출력한다. "contract.jsp"는 사용자가 [전자 계약] 메뉴를 선택할 때 서버에서 실행되며 원본 계약서 DB 테이블을 접근하여 원본 계약서의 리스트를 가진 HTML 코드를 웹 클라이언트로 전송한다. 사용자가 리스트에서 계약서를 선택하면 서버에서 "sign.jsp"가 실행되고 이 jsp는 클라이언트에서 M-ESign을 실행하도록 하는 <object> 태그를 포함하는 HTML 코드를 출력한다. 클라이언트에서는 M-ESign 코드가 실행되며 전자 서명한 후 즉시 서명된 계약서 데이타와 함께 "upload.jsp"를 실행하도록 HTTP 패킷을 서버로 전송한다. 서버는 "upload.jsp"를 실행하고 함께 전송된 서명된 계약서를 저장한다.

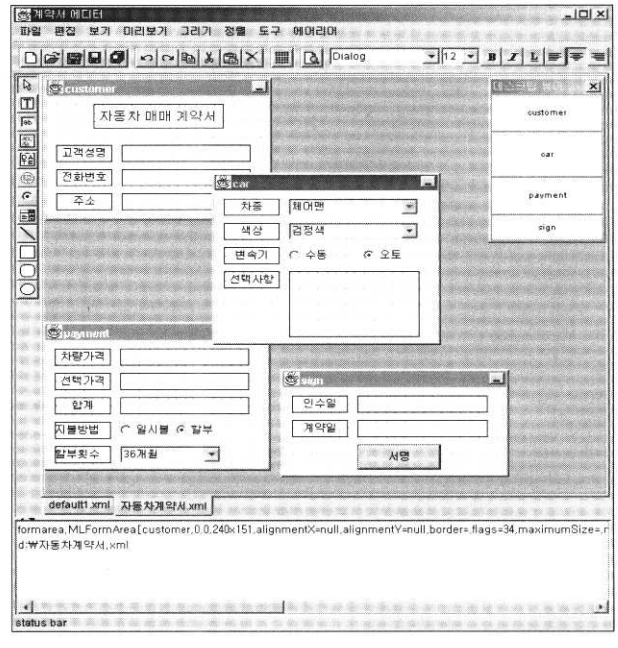
"index.jsp"가 전송한 HTML 코드가 출력된 웹 클라이언트에서 사용자가 [계약서 보기]를 선택하면 "login.jsp"가 서버에서 실행되며 웹 클라이언트 화면에 로그인 이름과 암호를 묻는다. 사용자가 로그인을 입력하면 서버에서 "list.jsp"가 실행되며 사용자를 고객 DB 테이블에서 확인한 후 성공하면 그 고객 소유의 서명된 계약서 리스트를 HTML 코드로 출력한다. 사용자가 임의의 계약서를 선택하면 "view.jsp"가 서버에서 실행되고 M-EDecoder 모듈을 실행하도록 하는 <object> 태그를 포함하는 HTML 코드를 출력한다. 웹 클라이언트에서 M-EDecoder가 실행되고 HTTP 요청 패킷을 전송하여 서명된 계약서를 서버로부터 다운로드 받아 서명을 확인한 뒤 이를 화면에 출력한다.



(그림 12) 전자 계약 서버의 JSP 페이지들의 상태 천이도

4.6 계약서 저작 도구(X-Auth)

계약서 저작 도구인 X-Auth 패키지는 GUI(Graphic User Interface)의 다양한 기능을 제공하는 JAVA 언어로 작성되었으며 JDK 1.3과 Swing을 이용하여 작성되었다. X-Auth의 사용자는 메뉴 아이콘을 이용하여 <표 1>에서 설명한 Area(영역)과 컨트롤 들을 임의로 생성할 수 있다. Area의 크기는 사용자가 미리 설정한 PDA 화면 픽셀 수에 따라 결정된다. 그러므로 한 화면 단위로 사용자는 계약서를 작성할 수 있다. 또한 사용자는 Area의 순서를 마음대로 결정할 수 있다. (그림 13)은 자동차 매매 계약서를 편집하는 X-Auth의 실행 화면을 보여 준다. 이 저작 도구를 이용하여 작성된 XML은 (그림 3)과 같다. X-Auth의 구체적인 구현에 관한 내용은 본 논문의 논점을 벗어나므로 설명을 생략한다.



(그림 13) X-Auth로 자동차 매매 계약서를 작성하는 실행 화면

5. 실행 및 성능 분석

5.1 실행 예

이 절에서는 본 논문에서 설계 구현한 시스템의 실행 예를 보이고 실행 시간의 성능을 측정한 내용을 기술한다. (그림 14)는 전자 계약 시스템을 실행한 화면들을 보여준다. (그림 14)(a)는 PDA로 웹을 이용하여 전자 계약 서버에 접속한 화면이다. 화면에서 [전자 계약] 메뉴를 선택하면 메뉴 리스트가 나타나고 그 때 "자동차 매매 계약서"를 선택하였다고 하자. (그림 14)(b)는 M-ESign 모듈이 자동차 매매 계약서 XML을 파싱한 후 윈도우 컨트롤을 이용하여 출력한 화면이다. M-ESign 모듈은 고객의 입력을 받고 전자 서명 후 서명된 계약서를 서버로 전송한다. (그림 14)(c)는 고객 컴퓨터의 M-EDecoder 모듈이 (그림 14)(b)에서 자신이 서명한 계약서를 검증한 후 출력한 화면이다.

(a) PDA에서 전자 계약 서버에 접속 (b) PDA에서 자동차 매매 계약서 출력 (c) 고객 자신이 작성한 계약서를 확인하는 화면

(그림 14) M-XContract의 실행 예

5.2 성 능

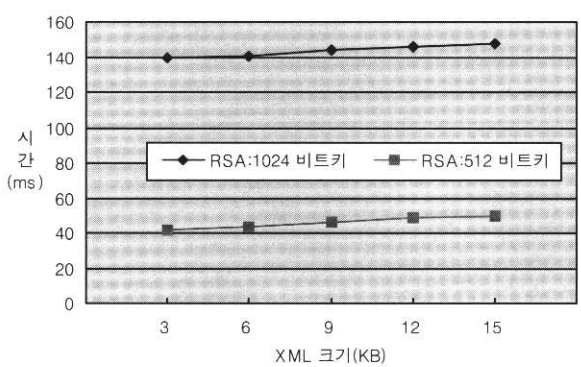
M-Commerce에 사용되는 단말기 및 이동 통신의 제약성으로 인해 모바일 시스템의 성능 평가시에는 단말기 처리 시간 및 이동 통신 전송에 소요되는 시간을 우선적으로 측정하는 것이 바람직하다. 따라서 본 논문에서는 M-XContract 시스템이 모바일 환경에서 적합한지를 평가하기 위해서 PDA에서 전자 서명에 걸리는 시간과 서명 데이터를 서버로 전송하는 데 걸리는 시간을 측정하였다.

본 실험의 환경은 <표 4>와 같이 구성되었다. 4~5개의 영역을 포함하는 계약서의 크기는 3K바이트 정도이기 때문에 3K바이트 단위로 계약서를 작성하였다. 서명 알고리즘으로 RSA를 사용한 이유는 무선 단말기 상에서 RSA 알고리즘의 적합성을 검사하기 위한 것이다. (그림 15)에 나타난 실험 결과는 총 20회 반복 실험의 평균 값이다.

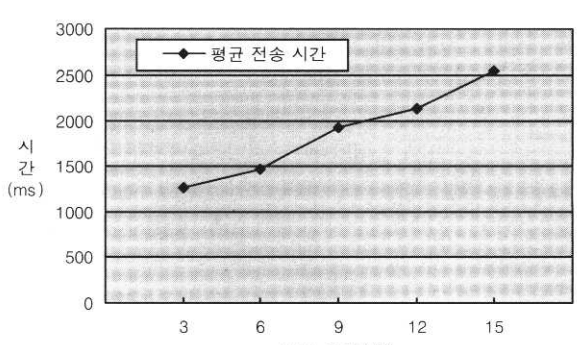
〈표 4〉 실험 환경

실 험 요 소	사 양
모바일 단말기	iPAQ 3850
해쉬 알고리즘	SHA-1
서명 알고리즘	RSA(512, 1024비트 키)
이동 통신 전송률	최대 144Kbps
XML 계약서 크기	3, 6, 9, 12, 15(KB)

(그림 15)(a)는 모바일 단말기에서 전자 서명을 생성하는데 걸리는 시간을 보여주고 있다. 키의 크기가 512비트인 경우에는 약 42~50ms가 소요되며 키의 크기가 1024비트인 경우에는 140~147ms가 소요되는 것을 확인할 수 있다. 일반적으로 RSA 서명 생성은 개인 키의 멱승을 포함하는데 개인 키가 클수록 서명 시간이 증가하게 된다. (그림 15)(b)는 모바일 단말기에서 무선으로 HTTP를 이용하여 서명 데이터를 서버로 전송하는데 소요되는 시간을 보여주고 있다. 서명 데이터에는 원본 메시지 외에 약 1200바이트가(인증서, 서명 값 등) 추가적으로 포함되었으며 전송 시간은 1257~2548ms 정도 소요되는 것으로 측정되었다.



(a) 전자 계약서에 대한 서명 생성 시간



(b) 서명된 계약서를 서버로 전송하는데 소요되는 시간

(그림 15) M-XContract 시스템의 성능

성능 평가의 결과를 요약하면 다음과 같다. 첫째, 전자 서명 생성 시간은 PDA와 같은 단말기에서 150ms 이내로서 사용자가 인내할 수 있는 시간이다. 둘째, 전자 서명 생성 시간은 계약서의 크기가 증가함에 따라 완만하게 증가

한다. 셋째, 전송 시간은 모바일 네트워크의 낮은 대역폭으로 인해 계약서의 크기와 밀접한 관련이 있다. 결론적으로 본 논문에서 제시한 XML 계약서는 호환성 및 다양한 사용자 인터페이스를 제공함과 동시에 계약서의 크기가 작아 모바일 환경에 적합하다고 할 수 있다.

6. 결 론

모바일 전자 계약이란 모바일 환경에서 고객이 전자 서명을 통해 매매를 계약하는 것을 말한다. 모바일 단말기의 하드웨어 한계나 단말기에 탑재된 시스템의 비호환성, 무선의 낮은 대역폭 등으로 인해 유선 상의 전자 계약 시스템을 M-Commerce에 그대로 적용하기에는 무리가 있다.

본 논문은 이러한 문제점들을 해결하기 위해 XML 기반의 전자 계약서를 정의하고 모바일 전자 계약 시스템 모델 및 M-XContract 시스템 구현 사례를 다루었다. M-XContract 시스템은 PDA를 모바일 단말기로 선택하였으며, 전자 계약 서버를 이용하여 원본 계약서와 서명된 계약서를 관리하게 하고, 고객은 언제든지 서버에 접속하여 자신의 계약서를 확인할 수 있게 하였다. 이를 위해 PDA 상에서 고객으로부터 계약 내용 입력 및 전자 서명 후 서버에 전송하는 M-ESign 모듈을 개발하고, 고객 컴퓨터에서 서명된 전자 계약서를 복호화하여 서명 확인 및 화면 출력을 실행하는 M-EDecoder를 개발하였다. 또한 PDA 화면에 적절한 크기로 출력될 수 있도록 전자 계약서를 XML 폼으로 정의하는 저작 도구 X-Auth를 구현하였다.

본 논문에서는 M-XContract 시스템의 효용성을 평가하기 위해 PDA 상에서 전자 서명에 걸리는 시간과 서명된 계약서를 서버로 전송하는 시간을 측정하였다. 성능 측정의 결과, RSA 알고리즘을 이용하였을 때도 전자 서명에 소요되는 시간은 1초에 훨씬 못 미치는 정도로서 사용자의 입장에서 전혀 무리가 없는 것으로 평가되었다. 또한 전자 서명 시간과 전송 시간은 계약서의 크기에 의존적이다. 마지막으로 모바일 네트워크를 통한 계약서의 전송 시간은 전자 서명 생성 시간에 비해 상대적으로 크고 계약서의 크기에 많은 영향을 받는다. 결론적으로 XML 기반 전자 계약서는 크기가 작고 다양한 인터페이스를 지원할 수 때문에 M-XContract 시스템은 모바일 환경에서 전자 계약 시스템을 위한 효과적인 모델로 평가할 수 있다.

향후에는 USB 토큰과 같은 인증서 저장 장치에 관한 연구를 수행함으로써 본 논문에서 제시한 전자 계약 시스템을 확장할 계획이다.

참 고 문 헌

[1] U. Varshney and R. Vetter, "A Framework for the Emerging Mobile Commerce Applications," Proceedings of the 34th Hawaii International Conference on System Sciences, 2001.

[2] James A. Senn, "The Emergence of M-Commerce," IEEE Computer, December, 2000.

[3] C. Bettstetter, H. Vogel and J. Eberspacher, "General Packet Radio Service GPRS : Architecture, Protocols, and Air Interface," IEEE Communications Surveys 2(3), 1999.

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

[5] M. Branchaud, "A Survey of Public-Key Infrastructures," Master's thesis, McGill University, 1997.

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

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

[8] R. Housley, Cryptographic Message Syntax, RFC 2630, Internet Society, 1999.

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

[10] M. Myers, R. Ankney, etc., Internet X.509 Public Key Infrastructure : Online Certificate Status Protocol-OCSP. RFC 2560, IETF, 1999.

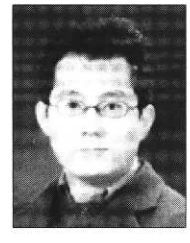
[11] 한국정보보호진흥원, 무선 PKI 기술 규격, http://www.kisa.or.kr, 2001.

[12] M. Heijden and M. Taylor, *Understanding WAP : Wireless Application, Devices, and Services*, Artech House, 2000.

[13] A. Jurisic and A. J. Menezes, "Elliptic Curves and Cryptography," Dobb's Journal, April, 1997.

[14] W3C, Extensible Markup Language(XML) 1.0. Technical report, WWW Consortium(W3C), http://www.w3.org/TR/1998/REC-xml-19980210, 1998.

[15] 전병선, *Microsoft Visual C++ 6.0 ATL COM Programming*, 삼양출판사, 1999.



황 기 태

e-mail : calafk@hansung.ac.kr

1986년 서울대학교 컴퓨터공학과(학사)

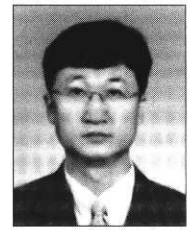
1988년 서울대학교 대학원 컴퓨터공학과 (공학석사)

1994년 서울대학교 대학원 컴퓨터공학과 (공학박사)

2000년~2001년 University of California, Irvine의 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 : 이동 통신 시스템, 정보 보안, 실시간 시스템 등