안전한 인터넷 전자지불 프로토콜의 설계 및 구현

박 현 동[†] · 이 은 성[†] · 송 상 헌[†] · 강 신 각[††] ·
박 정 수[†††] · 류 재 철[††††]

요 약

현재 인터넷에서 제공되고 있는 여러 응용 서비스 중에서 일반 사용자들이 가장 많이 이용하는 분야가 전자상거래이다. 그러나 현재 운영되고 있는 쇼핑서버들은 안전상의 문제점을 내포하고 있다. 쇼핑서버들은 나름대로의 방법으로 안전상의 문제점을 모두 해결하였다고 주장하지만 이러한 방법들이 전자지불 시스템에서 요구하는 보안 사항들을 모두 만족시켜 주지는 못한다. 본 논문에서는 현재 운영되는 쇼핑서버의 문제점들을 지적하고 이를 해결하는 안전한 전자지불 프로토콜인 SafePay를 제안하였고, 전자지갑을 기반으로 제안된 프로토콜을 구현한 결과를 소개한다.

Design and Implementation of a Secure Electronic Payment System on Internet

Hyun-Dong Park[†] · Eun-Sung Lee[†] · Sang-Heon Song[†] · Shin-Gak Kang[††] ·
Jeong-Soo Park[†††] · Jae-Cheol Ryou[††††]

ABSTRACT

One of the most popular services among the various application services provided in Internet is electronic commerce. However, it is true that the most of payment systems in existing shopping server are not secure enough even if they provide security services. In this paper, we pointed out some problems in existing shopping server. And, we designed and implemented the secure electronic payment system, SafePay to solve the problems in existing payment systems.

1. 서 론

인터넷 사용 인구의 증가와 사용 인구 계층의 일반화는 기업들의 인터넷 사업을 촉진시켰으며, 현재 세계의 유수 기업은 물론이고, 중소기업까지도 인터넷에 WWW 페이지를 개설하는 것을 기본적인 마케팅 수단으로 인식하게 되었다. WWW 페이지를 이용하여 단지 자사를 광고하던 단계에서 벗어나 이제는 기업과 소비자들이 인터넷을 통하여 직접 상품의 거래를 수행하고 있다. 전자 상거래는 여러 가지의 형태로 서비스된다. 기업간의 상거래와 기업과 소비자간의 상거래 등의 형태가 있지만, 일반 사용자들이 이용하게 되는 것은 주로 백화점이나 유통업체의 WWW 페이지를 이용하여 상품을 구입하는 소매 형태의 상거래이다[1].

인터넷을 통한 상거래는 기업과 소비자 모두 기존의 시공 개념을 초월하기 때문에 전자 상거래의 규모는 앞으로 크게 확장될 것이라고 예상된다. <표 1>은 세계와 국내의 전자 상거래 시장 규모를 보여 준다[2]. <표 1>에서 볼 수 있듯이 전자 상거래의 규모는 매년

† 준 회 원 : 충남대학교 대학원 컴퓨터과학과
†† 정 회 원 : 한국전자통신연구원 표준연구센터 표준기획연구팀 책임연구원
††† 정 회 원 : 한국전자통신연구원 표준연구센터 연동표준연구팀 연구원
†††† 종신회원 : 충남대학교 컴퓨터과학과 교수
논문접수 : 1999년 5월 4일, 심사완료 : 1999년 7월 22일

2배 이상으로 성장하고 있으며, 앞으로 지구상에서 발생하는 총 교역량의 20% 정도가 인터넷을 통해 이루어질 것으로 전망되고 있다.

〈표 1〉 전자상거래의 규모

연 도 / 지 역	1996	1997	1998	1999	2000
세계 (백만불)	518	1,188	2,371	2,990	6,570
국내 (백만원)	1,400	6,285	15,004	34,484	61,396

* 출처 : 한국정보통신진흥협회, 정보화사회, 1998년 3, 4월호

하지만, 이러한 전자 상거래가 장점만을 갖추고 있는 것은 아니다. 이러한 장점들의 이면에는 극복해야 할 여러 가지 문제점들도 많이 내재되어 있다. 가장 민감한 문제점은 경제 정보에 대한 보안이다. 전자 상거래 초기에는 주문은 인터넷에서 수행하고 대금의 지불은 은행의 지로를 이용하는 등의 방법을 이용해 왔다. 그러나 전자 상거래가 발전하면서 주문 이외에 인터넷을 통해 대금의 지불까지도 수행하는 형태가 일반적인 형태로 자리 잡아 가고 있다. 현재 개설되어 있는 거의 모든 쇼핑서버에서도 대금 지불까지 수행해 주는 형태이다[3][4]. 현재 운영되는 쇼핑서버에서 가장 많이 제공해 주는 지불 방법은 SSL을 이용하여 신용카드 정보를 전송하는 방법과 무통장 입금 방법이다. 무통장 입금 방법은 인터넷에서 지불이 처리되지 않기 때문에 전자 지불 방법으로 고려하지 않았다. SSL을 이용하여 신용카드 정보를 전송하는 방법도 여러 가지의 문제점을 내포하고 있다. 신용카드 이외에도 전자화폐 등을 이용할 수 있지만 전자화폐는 아직까지 기술적인 문제가 남아 있고, 보안 개념이 미비한 일반 사용자들의 사용은 시기상조라는 것이 일반적인 견해이다[5]. 본 논문에서는 신용카드를 이용하는 전자 지불 프로토콜인 SafePay를 설계하였다. 또한 사용자에게 보다 현실과 유사한 인터페이스를 제공하기 위해 전자지갑을 구현하여 SafePay의 구현에 이용하였다. SafePay는 현재 운영중인 쇼핑서버에서 문제점으로 지적되는 모든 보안 요구사항을 만족시켜 주는 전자 지불 프로토콜이다.

본 논문의 2장에서는 SET을 비롯한 사용되고 있는 전자지불의 방법들과 이러한 방법들이 가지고 있는 약점을 알아 본다. 3장에서는 본 논문에서 제안한 SafePay의 동작 원리와 메시지의 구성 형태를 설명한다. 4장에서는 SafePay의 구현에 관련된 사항을 알아 보고, 5장에서는 안전성을 살펴 본다. 마지막으로 6장에서 결론과 앞으로 연구되어야 할 사항들을 알아 본다.

2. 전자지불 방법과 문제점

현재 국내에서 운영되고 있는 쇼핑서버들은 모두 자신들이 제공하는 지불 방법이 가장 안전하다고 주장하고 있지만, 사실은 전혀 안전하지 않은 상황에서 운영되고 있다. 2장에서는 현재 전자지불 시스템의 대표적인 방식으로 평가되고 있는 SET의 구성과 문제점에 대해서 알아 본다. 그리고, 국내의 쇼핑서버들에 의해 제공되는 지불 방법들의 유형을 알아 보고, 각각이 가지고 있는 약점을 통해 어떠한 공격이 가능한 지를 알아 본다.

2.1 SET(Secure Electronic Transaction)

SET은 신용카드 회사인 VISA와 Master Card 사가 신용카드를 기반으로 인터넷 상의 전자결제를 안전하게 이룰 수 있도록 마련한 전자결재과정 표준안이다[6]. SET이 전자 상거래에서 요구하는 사항들을 모두 만족시켜 줄 수 있지만, 이를 구현하여 실제로 응용하는 데에는 많은 문제점이 있다. 첫째는 인증기관의 구축이다. SET에서는 인증서의 구성을 X.509에 기초하고 있다. X.509가 안전한 인증서의 사용을 제공하고 있지만, 시스템의 복잡성 때문에 전자 상거래의 주체인 일반 사용자들이 이를 사용하는 데에는 한계가 있다. 두번째는 시스템의 복잡성이다. 전자 상거래의 규모는 다양하다. 큰 금액이 다루어지는 거래도 있지만, 인터넷의 쇼핑몰에서 이루어지는 거래는 금액이 크지 않은 것들이다. 작은 금액의 거래에 많은 기관이 참여하게 되면 그곳에서 발생하는 수수료도 증가하여 거래 금액에서 많은 부분을 차지하게 된다. 이 밖에도 미국과 우리의 금융환경이 다르다는 것도 문제점으로 지적된다. 결국, SET은 이론적으로는 안전한 전자지불을 수행해 줄 수 있지만, 실제로 이를 사용하기에는 극복하기 어려운 문제점을 남겨 놓고 있다. 본 논문에서 제안하는 SafePay는 SET에서 주장하는 보안사항을 모두 만족시키면서 시스템의 크기를 줄여 놓은 시스템으로 금융기관과의 합의만 이루어지면 언제든지 실제의 시장에서 이용될 수 있는 장점이 있다.

2.2 전자지불 방법

쇼핑 서비스와 함께 제공되는 지불 방법에는 여러 가지의 방법이 있다. 그러나, 현재 운영중인 쇼핑서버에서는 2~3개의 방법만을 사용하고 있다. 이 방법들을 알아 보면 다음과 같다[3][4].

□ **온라인 입금**

WWW 페이지에 쇼핑서버의 계좌를 표시하고, 고객으로 하여금 온라인으로 입금을 하게 하는 방법이다. 가장 간단하며 보안 요구사항이 비교적 적은 방법이다. 그러나, 지불 행위가 네트워크 상에서 발생하지 않고 전통적인 방법을 사용하고 있으므로, 전자 지불이라고 할 수 없는 방법이다.

□ **SSL과 폼페이지를 이용한 신용카드 정보 전송**

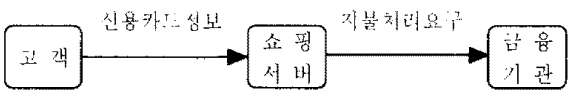
모든 쇼핑서버에서 제공해 주는 방법이다. 지불 처리 단계에서 쇼핑서버는 폼페이지를 고객의 브라우저로 전송하는 데, 이 폼페이지에 고객의 신용카드 정보를 입력케 하는 방법이다. 고객이 입력한 신용카드 정보는 SSL을 이용하여 암호화되어 쇼핑서버에게 전송된다.

□ **전자지갑을 이용한 신용카드 정보 전송**

기본적인 개념은 폼페이지를 이용하는 방법과 동일한 원리이지만 쇼핑서버가 전송한 폼페이지에 정보를 입력하지 않고, 고객의 컴퓨터에 설치되어 있는 전자지갑 프로그램에 정보를 입력한다는 것이 차이점이다. 고객의 전자지갑에 입력된 신용카드 정보는 HTTP 또는 별도의 소켓을 이용하여 쇼핑서버에게로 전송된다. HTTP를 이용할 때에는 SSL을 이용하고, 소켓을 이용할 때에는 별도의 암호 기술을 이용한다.

2.3 전자지불의 약점

현재 운영되고 있는 전자지불 방법들의 정보 흐름을 보면 (그림 1)과 같다.



(그림 1) 전자지불정보 흐름도

(그림 1)과 같은 구성에서 발생할 수 있는 문제점을 알아 보면 다음과 같은 것들이 있다.

□ **신용카드 정보의 노출과 저장**

쇼핑서버들은 SSL을 사용하여 암호화된 메시지를 주고 받기 때문에 보안상의 위험이 전혀 없다고 한다. SSL을 사용한다면 WWW 통신 자체에는 아무런 문제가 없을 것이다[7]. 그러나, 쇼핑서버는 모든 신용카드 정보를 볼 수 있고 또한 저장해 놓을 수 있다. 쇼핑서버가 고객의 신용카드 정보를 저장해 놓는 것은 실세계에서 고객의 신용카드를 상점에 맡겨 놓는 행위와 비교할 수 있다. 상점에서 신용카드를 이용하여 상품을 구입한다고 해서 신용카드를 맡겨 놓을 필요는 없듯이, 쇼핑서버가 고객의 신용카드 정보를 볼 수 없어야 하고, 저장해 놓을 수도 없어야 한다.

□ **지불처리의 주체**

A의 통장에서 B의 통장으로 돈이 이동될 때는 그 거래의 요청을 A가 하는 것이 일반적이다[4]. 그러나, 현재의 상황에서는 금융 기관에 대금 이체를 요청하는 쪽은 쇼핑서버, 즉 돈을 받는 당사자이다. 고객이 자신의 신용카드 정보를 입력하였기 때문에 거래의 시작은 고객에서부터 출발하였다고 할 수 있으나, 앞에서 살펴 본 것처럼 쇼핑서버가 고객들의 신용카드 정보를 저장할 수 있고, 전자서명의 사용이 배제된 상황에서는 고객의 요구 없이 지불을 신청할 수 있다. 고객이 직접 대금 이체를 요구할 수 있는 환경이 요구되는데, 이의 구현을 위해서는 전자서명의 사용과 거래를 관리하는 제 3의 기관 설립이 필수적이다.

□ **영수증 부재**

모든 거래에 있어 그 거래의 증거 자료로 이용할 수 있는 것이 영수증이다. 영수증은 언제 누가 누구에게서 얼마의 가격에 어떤 상품을 구입했는지를 증명해 줄 수 있는 문서이다[4]. 만약 사용자가 대금을 지불한 후, 상품을 전달 받지 못하면 영수증을 근거로 하여 판매자에게 이의를 제기할 수 있다. 그러나, 현재 운영중인 쇼핑서버들 중에 영수증을 발급해 주는 곳은 없다. 상품의 구입이 완료된 후에 고객에게 전송되는 것은 구입 결과를 알려 주는 전자우편 정도이다. 그러나, 쇼핑서버가 상품을 전달하지 않는 등의 분쟁이 발생했을 때, 이 전자 우편 메시지를 어느 정도로 이용할 수 있는 지는 의문이다. 그 이유는 전자 우편 메시지에 전자서명이 없기 때문이다.

□ **장바구니 기능의 역효과**

인터넷 쇼핑서버의 수가 증가하기 시작하면서 사업

자들은 고객들에게 좀 더 현실 세계와 가까운 환경을 제공하고자 노력하였다. 그 결과로 나온 기능 중의 하나가 장바구니 기능이다. 장바구니 기능을 구현하는 방법은 장바구니의 정보가 저장되는 위치를 기준으로 하여 두 가지의 형태로 구분된다. 하나는 쇼핑서버에 위치하는 것이고, 다른 하나는 사용자의 컴퓨터에 위치하는 것이다.

장바구니 정보를 고객의 컴퓨터에서 저장하는 경우에 발생할 수 있는 문제는 상품정보의 변경이다. 이를 방지하기 위해 장바구니 기능을 쇼핑서버에서 관리하는 방법으로 구현된 쇼핑서버들도 있다. 이러한 환경에서는 쇼핑서버에 의한 상품정보 변경도 고려해야 한다. 이 방법에서 생각할 수 있는 또 하나의 위험요소는 다른 사용자의 장바구니 사용이 가능하다는 점이다. 이것이 가능한 이유는 HTTP가 한 쌍의 메시지로 구성되고, 한 쌍의 통신이 종료되면 다음의 통신은 이전의 통신과 아무런 연관성을 가지지 않기 때문이다[7][8].

□ SSL의 취약성

미국 이외의 나라에서 사용할 수 있는 SSL은 40비트 크기의 암호키를 이용하여 메시지를 암호화한다[9]. 그런데, 40비트의 키로 암호화된 암호문은 암호키 정보 없이 짧은 시간 안에 복호화될 수 있다. RSA사에서 세번째로 실시한 DES Challenge III에서 12시간 15분만에 암호화에 사용된 키를 찾아내었다[5]. 이 실험에는 Deep Crack이라는 칩을 설치한 DES Cracker라는 전용 하드웨어를 사용하였다. 이러한 실험이 보여주듯이 40비트의 키를 이용하여 신용카드 정보처럼 중요한 정보에 대해 기밀성을 보장해 주는 것은 무리이다. 그러므로, SSL을 이용하여 사용자의 정보를 비밀스럽게 전송한다는 쇼핑서버의 주장은 설득력이 없다.

앞에서 살펴본 바와 같이, 현재 운영되는 쇼핑서버들이 제공하는 지불 방법들은 모두 심각한 취약점을 가지고 있다. 이러한 취약점을 포함한 상태로는 전자상거래의 발전을 더 이상 기대할 수 없으며, 이 상황을 방치할 경우에는 큰 혼란이 예상되므로 이러한 취약점을 모두 극복하는 전자지불 프로토콜의 개발이 시급하다.

3. 안전한 전자지불 프로토콜 SafePay

SafePay는 인터넷의 WWW 서버로 운영되는 쇼핑서버에서 사용할 수 있도록 설계된 전자지불 프로토콜이다. 2장에서 설명한 현재 운영 중인 쇼핑서버들이 가지고 있는 모든 문제점들을 해결할 수 있도록 설계되었다. 사용자 인터페이스는 전자지갑을 채택하였다. 전자지갑을 사용하는 것이 사용자들에게 가장 편리한 환경을 제공하고, 다양한 지불 프로토콜을 구성할 수 있기 때문이다.

3.1 SafePay의 구조

SafePay는 크게 보면 5개의 구성원으로 이루어진다. 각 구성원들에 대한 설명은 다음과 같다.

□ **고 객** : 고객은 상품을 구입하면서 대금을 지불하는 거래 당사자이다. 신용카드를 사용하여 지불을 수행하며 신용카드의 비밀번호를 알고 있어야 대금을 지불할 수 있다. 자신의 공개키/비밀키 쌍을 생성하여 인증서버로부터 인증서를 발급받는다.

□ **쇼핑서버** : 대금을 받고 고객에게 상품을 공급하는 거래 당사자이다. 고객과 마찬가지로 공개키/비밀키 쌍을 생성하여 인증서버로부터 인증서를 발급받는다. 쇼핑서버는 고객의 신용카드에 관한 정보를 알아낼 수 없다.

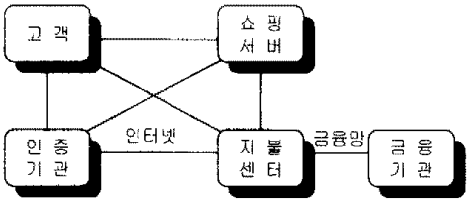
□ **지불센터** : 고객과 쇼핑서버 사이에서 거래와 대금의 이동을 담당한다. 지불센터는 고객과 쇼핑서버로부터 신뢰를 받는 기관으로써 고객과 쇼핑서버 사이에서 발생하는 모든 거래에 대해 정당성을 판별하고, 발생하는 모든 분쟁에 대한 책임을 갖는다. 발생하는 모든 거래에 대한 정당성을 판단해 주고, 지불을 처리해 주기 때문에 지불센터의 역할이 가장 중요하다. 그러므로, SafePay의 구현에 있어서는 고객과 쇼핑서버가 지불센터에게 지불해야 할 수수료 문제도 언급되어야 하지만, 본 논문에서는 이 사항에 대해 고려하지 않았다.

□ **인증서버** : SafePay 구성원들의 공개키를 받아 인증서를 발급하는 역할을 담당한다.

□ **금융 기관** : 지불센터로부터 고객과 쇼핑서버의 정보를 수신하여 실제의 대금 이동을 수행하는 기관이다. 지불센터와 통신하며, 이외의 다른 구성원들과는 연결되어 있지 않다.

이들 구성원들의 구조를 그림으로 보면 (그림 2)와

한다.

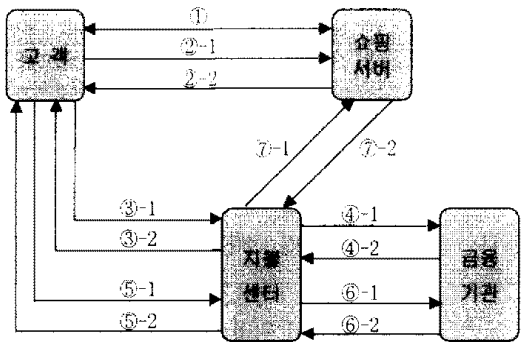


(그림 2) SafePay의 구성원

모든 구성원들은 인터넷에 연결되어 운영된다는 전제에서 출발하였지만, 은행, 신용카드 회사 등의 금융기관은 아직 업무 처리를 인터넷으로는 수행하지 않으므로 SafePay에서는 인터넷을 이용한 금융 기관과의 통신은 고려하지 않았고, 지불센터와 금융기관 사이의 통신은 기존의 금융망을 사용하는 것이라고 가정하였다. SafePay에 지불센터를 별도로 구축한 것은 쇼핑서버에서 지불 업무까지 수행하는 현재의 시스템들이 큰 보안 문제점을 가지고 있기 때문이다. 지불센터는 고객이나 쇼핑서버와는 이해관계가 없는 중립적인 기관으로써 고객과 쇼핑서버 모두에게 신뢰를 받고 있다는 가정에서 출발한다. 만약, 지불센터에서 의도적이거나 혹은 실수로 인하여 지불에 문제가 발생한다면 이 지불센터를 이용하여 지불을 수행하는 고객과 쇼핑서버의 수가 줄어들 것이므로 해당 지불센터는 자연스럽게 도태될 것이다. 즉, 이러한 문제는 시장 경제의 논리로 풀어야 할 문제이다.

3.2 SafePay의 메시지 흐름

SafePay의 각 메시지가 어떠한 형태로 암호화되며, 그 암호 동작이 어떤 기능의 구현을 위해서인지를 설명한다. SafePay를 구성하는 모든 메시지들의 구성을 그림으로 보면 (그림 3)과 같다.



(그림 3) SafePay의 메시지들

모든 메시지들이 번호의 순서대로 수행되는 것은 아니다. 예를 들어 ③-2번의 경우, ④-1과 ④-2의 수행이 완료된 후에 수행된다. 각 메시지들의 암호문 형태와 그 내용을 자세히 살펴 보기에 앞서 프로토콜에 사용된 암호 기호들의 의미를 설명하면 다음과 같다.

KR(KU)x : 사용자 x의 비밀(공개)키
KS# : 새로 생성된 세션키
EKR(KU)x : 사용자 x의 비밀(공개)키를 이용한 암호화
DKR(KU)x : 사용자 x의 비밀(공개)키를 이용한 복호화
H(M) : M의 해쉬함수 결과값

□ 쇼핑 단계(①)

쇼핑서버에 접속하여 상품 구입 버튼을 클릭하면 상품의 정보가 고객의 컴퓨터로 전송되면서 설치되어 있는 전자지갑의 장바구니에 저장된다. 이 부분에서 플러그-인을 이용하여 전자지갑을 구동하게 된다. 이 부분에 대한 자세한 설명은 구현 부분에서 한다.

□ 주문 메시지(②-1)

E KR 고객 [H(OI)] || E KS1 [OI] || E KU 쇼핑서버 [KS1] || 고객인증서

OI는 Order Information의 약자로써, 상품정보를 나타낸다. 즉, 고객이 이번 거래에서 구입하려고 하는 상품에 관한 정보들을 나타낸다. OI를 구성하는 메시지의 모습은 (그림 4)와 같다.

```
Md5039 : 유닉스 헤어드라이어 : 32,000 : 3
USTW200 : LG가정용청소기 : 120,000 : 1
216,000
hdpark-0011
대전시 유성구 충남대학교 분산처리연구실
```

(그림 4) OI의 내용

OI에는 상품의 모델기호와 상품명, 주문 개수, 상품별 단가, 총 금액, 거래ID, 배달지가 포함된다. (그림 4)의 위 두 줄이 구입 상품에 관한 정보이다. "Md5039"는 모델기호이고, "유닉스 헤어드라이어"는 상품명, "32,000"은 상품의 단가, "3"은 주문 개수를 각각 나타낸다. 세번째 줄의 "216,000"은 구입하려는 두 종류 네 가지 상품의 총 금액이다. "hdpark-0011"은 거래ID로써, "hdpark"은 고객의 고유 ID이고, "0011"은 "hdpark"이라는 ID를 가지는 사용자가 이 쇼핑서버에 11번째로 주문하는 거래라는 사실을 나타낸다. 거래ID는 쇼핑서

버에서도 고객별로 저장되므로, 이를 이용하여 재전송 공격을 방지할 수 있다. 마지막으로 상품이 배달될 주소가 포함된다.

② 1 메시지의 암호문 구성을 보면 OI를 쇼핑서버만이 복호화할 수 있도록 암호화하고(E KS1 [OI] || E KU 쇼핑서버 [KS1]), OI에 대한 고객의 전자서명을 추가한다(E KR 고객 [H(OI)]). 여기에서 얻을 수 있는 기능은 고객의 사생활 보장과 고객에 의한 주문 사실 부인, 가격 부인에 대한 봉쇄이다. 고객이 어떤 상품을 구입하는 지를 쇼핑서버만이 알 수 있고, 지불센터는 상품의 종류를 알 수 없다. 또한 메시지에 고객의 서명이 포함되기 때문에 후에 고객이 주문 사실 자체를 부인하거나, 가격에 대해 오류를 주장할 때 고객의 주장을 반박할 수 있는 증거로 사용된다[10].

고객인증서는 고객의 공개키에 대해 인증서버가 발행한 인증서이다. SafePay에서는 고객의 인증서를 ②-1 메시지를 전송할 때마다 포함시키도록 되어 있다. 이렇게 함으로써 인증서 관리 부분을 더욱 간소화할 수 있다. 쇼핑서버와 지불센터의 인증서는 고객이 주문 이전에 소유하고 있어야 하는 것이며, 쇼핑서버로부터 다운로드하는 전자지갑 프로그램에 인증서와 공개키 화일을 포함시킨다. 이에 대한 자세한 설명은 구현 부분에서 한다.

□ **가격 합의 메시지(②-2)**

E KR 쇼핑서버 [H(OI)] || E KR 쇼핑서버 [H(총액 || 거래ID || 쇼핑서버ID)] || E KU 지불센터 [총액 || 거래ID || 쇼핑서버ID]

②-1 메시지를 수신한 쇼핑서버는 전자서명을 확인하고, OI의 내용을 확인한다. 이 때에 가장 중요한 부분은 고객이 작성한 총액이 올바르게 계산되었는 지를 확인하는 것이다.

고객이 전송한 메시지가 모두 정당한 것으로 확인되면, 쇼핑서버는 고객이 전송한 정보를 확인해 주는 전자서명 메시지(E KR 쇼핑서버 [H(OI)])와 지불센터에 자신의 신원을 밝힐 수 있는 메시지(E KR 쇼핑서버 [H(총액 || 거래ID || 쇼핑서버ID)] || E KU 지불센터 [총액 || 거래ID || 쇼핑서버ID])를 구성하여 고객에게 전송한다.

"E KR 쇼핑서버 [H(OI)]"는 고객이 전송한 OI에 대한 쇼핑서버의 전자서명을 생성한 것으로 "E KR 고객 [H(OI)]"와 함께 이용되어 고객과 쇼핑서버가 상품정보 및 가격에 동의했다는 증거로 사용할 수 있다[10]. "E KR 쇼핑서버 [H(총액 || 거래ID || 쇼핑서버ID)]"는 고객에 의해 지불센터로 전송될 메시지로 이번 거래를 쇼핑서버가 정당한 것으로 인정한다는 의미이다. 그리고, "E KU 지불센터 [총액 || 거래ID || 쇼핑서버ID]"는 지불센터만이 복호화할 수 있도록 암호화한 것이다. 여기에서 관용 암호 알고리즘을 사용하지 않은 이유는 암호화가 수행되는 메시지인 "총액 || 거래ID || 쇼핑서버ID"의 길이가 짧기 때문에 공개키 암호 알고리즘을 사용하더라도 수행시간에 큰 영향을 주지 않기 때문이다. 결국, 관용 암호 알고리즘에 소요되는 시간만큼 수행시간을 단축시킬 수 있다.

②-1과 ②-2 메시지의 전송으로 고객과 쇼핑서버가 상품과 관련한 모든 사항에 대해 합의를 했다는 증거를 남기게 된다. 쇼핑서버는 OI 정보를 자신의 컴퓨터에 저장하여, 후에 지불센터가 "거래 성사 확인" 메시지를 전송하면 이 정보를 근거로 상품을 배달하게 된다.

□ **지불 요구 메시지(③-1)**

H(OI) || H(PI) || E KR 고객 [H(OI)] || E KR 쇼핑서버 [H(OI)] ||

E KR 쇼핑서버 [H(총액 || 거래ID || 쇼핑서버ID)] || E KU 지불센터 [총액 || 거래ID || 쇼핑서버ID] ||

E KR 고객 [H(PI)] || E KS2 [PI] || E KU 지불센터 [KS2]

쇼핑서버로부터 ②-2 메시지를 수신한 고객은 쇼핑서버의 전자서명 부분인 "E KR 쇼핑서버 [H(OI)]"를 복호화함으로써 쇼핑서버가 동일한 상품정보에 대해 합의했다는 사실을 확인할 수 있다. 이 확인작업이 수행되면 지불센터로 전송할 메시지를 구성하는 데, ②-2에서 수신한 정보가 이용된다.

"H(OI) || H(PI) || E KR 고객 [H(OI)] || E KR 쇼핑서버 [H(OI)]"부분은 동일한 상품정보에 대해 고객과 쇼핑서버가 전자서명을 생성함으로써 고객과 쇼핑서버가 상품에 대해 합의를 했다는 증거로 삼는다. PI는 "Payment Information"의 약자로서 지불정보, 즉 고객의 신용카드 정보를 의미한다. PI를 구성하는 내용은 (그림 5)와 같다.

BC카드 : 9410283962910012 : 3809 : 216,000 : hdpark-0011

(그림 5) PI의 내용

PI를 구성하는 내용에는 먼저 고객이 사용하려고 하

는 신용카드의 정보가 포함된다. "9410283062910012"는 신용카드의 번호이고, "3809"는 신용카드의 비밀번호이다. "216,000"은 이번 거래의 총 금액을 나타내고, "hd park-0011"은 거래ID를 의미한다. PI는 고객이 지불센터 이외의 사용자들에게는 노출을 원하지 않는 내용들이다.

"E KR 쇼핑서버 [H(총액 || 거래ID || 쇼핑서버ID)] || E KU 지불센터 [총액 || 거래ID || 쇼핑서버ID]" 부분은 ②-2에서 고객이 쇼핑서버로부터 수신한 내용이다. 고객은 이 메시지를 수신하여 지불센터로 전송할 뿐, 이 메시지에 대해서는 어떠한 작업도 수행하지 않는다. 이 메시지에는 쇼핑서버의 전자서명이 포함되어 있기 때문에 고객이 총액의 내용을 임의로 변경할 수 없도록 하는 기능을 제공한다[6].

"E KR 고객 [H(PI)] || E KS2 [PI] || E KU 지불센터 [KS2]" 부분은 PI의 내용에 대해 고객이 전자서명을 계산하고, PI의 내용을 지불센터만이 복호화할 수 있도록 암호화한 부분이다. 이러한 암호화 절차를 통해, 지불센터는 PI를 고객이 스스로 구성하였다는 사실을 확인할 수 있고, 고객은 PI를 지불센터만이 볼 수 있도록 암호화하여 전송한다.

③-1 메시지는 SafePay 프로토콜에서 가장 많은 정보가 전송되고, 또한 가장 중요한 정보들이 전송되는 메시지이다. 이 메시지에는 고객과 쇼핑서버가 OI 및 총액에 대해 합의를 했다는 증거로 사용할 수 있는 내용이 포함되어 있다. 발생할 수 있는 대부분의 분쟁에 대해 지불센터는 이 메시지에 포함되어 있는 내용을 이용하여 분쟁을 해결할 수 있다.

□ 신용카드 인증 요구 메시지(④-1)

③-1 메시지에서 H(OI) 부분은 후에 고객에게 영수증을 발급할 때와 쇼핑서버로 배달을 명령할 때 사용된다. 동일한 OI에 대해 고객과 쇼핑서버가 각각 "E KR 고객 [H(OI)] || E KR 쇼핑서버 [H(OI)]"의 형태로 전자서명을 생성해 놓았기 때문에 지불센터는 이 두개의 전자서명이 일치하는 지를 확인한다. 만약, 두 개의 전자서명을 복호화하여 생성된 두 개의 H(OI)가 일치하지 않을 경우에 지불센터는 ③-2 메시지에서 지불 취소 메시지를 전송한다. 또한, PI에 포함되어 있는 총액, 거래ID와 쇼핑서버가 계산한 "E KR 쇼핑서버 [H(총액 || 거래ID || 쇼핑서버ID)] || E KU 지불센터 [총액 || 거래ID || 쇼핑서버ID]"에 포함되어 있는 총액, 거래ID가 일치하지 않을 경우에도 취소 메시지를 전송한다.

모든 정보들이 정당한 것으로 판단되면, ③-2 메시지가 전송되기 전에 지불센터는 ④-1과 ④-2 메시지를 먼저 수행한다. ④-1은 ③-1 메시지를 수신한 지불센터가 금융 기관으로 신용카드의 인증을 요구하는 단계이다. SafePay에서는 이 단계의 구현은 고려하지 않았다.

□ 신용카드 인증 응답 메시지(④-2)

④-1 메시지를 수신한 금융 기관이 신용카드의 인증 작업을 수행한 후, 그 결과를 지불센터로 통보해 주는 단계이다. 이 메시지의 내용으로 다음에 전송될 ③-2 메시지의 내용이 결정된다. 이 단계의 구현도 SafePay에서는 고려하지 않았다.

□ 영수증 발급 메시지(③-2)

E KR 지불센터 [H(H(OI) || H(PI))]

④-2 메시지에서 신용카드 인증 결과를 수신한 지불센터는 신용카드가 지불 가능한 카드로 확인되면, 고객이 영수증으로 이용할 수 있는 메시지를 구성하여 고객에게 전송한다. 여기에서 중요한 점은 지불센터가 금융 기관으로 아직 지불 명령을 내리지 않았다는 것이다. 현재까지는 신용카드의 사용 가능 여부만 확인한 것 뿐이다. 이는 지불센터에서 발급한 영수증에 대해 고객이 이의를 제기하여 거래를 취소하는 상황이 발생할 수 있기 때문이다[3][5].

영수증의 내용은 ③-1 메시지에서 수신한 "H(OI) || H(PI)"를 다시 한번 해쉬함수에 입력하여 계산된 "H(H(OI) || H(PI))"에 대해 지불센터의 전자서명을 계산한 것이다. H(OI)와 H(PI)는 고객이 모두 소유하고 있는 정보이므로 고객이 이의 내용을 확인할 수 있다.

□ 영수증 수신 확인 메시지(⑤-1)

E KR 고객 [H(H(H(OI) || H(PI)))]

③-2 메시지를 수신한 고객은 자신이 소유하고 있는 H(OI)와 H(PI)를 이용하여 영수증의 전자서명을 확인한다. 영수증의 내용이 일치하면, 고객은 영수증을 올바르게 수신했다는 의미로 "H(H(OI) || H(PI))"를 다시 한번 해쉬함수의 입력으로 하여 "H(H(H(OI) || H(PI)))"를 계산한 후, 이에 자신의 전자서명을 계산하여 지불센터에게 전송한다. 지불센터가 전송한 "H(H(OI) || H(PI))"를 다시 해쉬함수로 계산하는 이유는 이 메시지를 포함한 이후의 메시지에 대한 재전송 공격을 막기 위함이다.

이 단계에서 영수증이 고객의 전자지갑에 저장된다. 하지만, 이 영수증이 제 역할을 다 하지는 않는다. 다음 단계에서 처리되는 "지불 처리 요구 메시지"와 "지불 처리 응답 메시지" 과정이 수행된 후, "지불 처리 통보 메시지"를 수신하면 전자지갑에 저장된 영수증이 역할을 수행할 수 있다.

□ **지불 처리 요구 메시지(⑥-1)**

고객이 영수증을 올바르게 수신했고 영수증의 내용에 대해 이의가 없다는 의미인 ⑤-1 메시지를 수신한 지불센터는 금융 기관으로 지불 처리를 요구하는 메시지를 전송한다. 이때는 ③-1 메시지에 포함되어 있던 PI 정보를 이용한다. SafePay에서 이 단계는 고려하지 않았다.

□ **지불 처리 응답 메시지(⑥-2)**

앞선 ⑥-1 메시지에 대한 금융 기관의 응답 메시지이다. 지불 처리가 정상적으로 수행되었을 때와 그렇지 않은 경우, 지불센터로 전송되는 메시지의 내용이 다르게 된다. SafePay에서 이 단계는 고려하지 않았다.

□ **지불 처리 통보 메시지(⑤-2)**

E KR 지불센터 **[H(H(H(H(OI) || H(PI))))]**

금융 기관으로부터 ⑥-2 메시지를 수신한 지불센터는 고객에게 ③-2 메시지에서 수신한 영수증이 제 기능을 다할 수 있도록 해주어야 한다. 고객이 전송한 ⑤-1 메시지에 포함되어 있던 "H(H(H(OI) || H(PI)))"의 해쉬함수값을 다시 구한 다음, 이에 지불센터의 전자서명을 계산하여 이를 전송한다. 이 메시지는 고객과 지불센터 사이에서 발생하는 마지막 메시지로써 고객이 이 메시지를 수신해야 앞서 수신했던 영수증이 제 역할을 모두 수행할 수 있게 된다. 이 메시지를 수신하기 이전까지는 단순히 영수증이 화일에 저장된 상태이다. 이 메시지를 수신하면 정상적인 거래라는 표시를 영수증이 저장된 화일에 부착하여 후에 정당한 영수증으로 사용할 수 있게 한다.

□ **배달 요구 메시지(⑦-1)**

E KU 쇼핑서버 **[거래ID] || E KR** 지불센터 **[H (OI)]**

지불센터는 고객과의 통신, 금융 기관과의 모든 통신을 종료한 후에 쇼핑서버로 하여금 고객에게 상품을 배달하라는 지시를 내려야 한다. "E KU 쇼핑서버 [거래ID]"를 전송함으로써 어느 거래에 대한 배달 요구인지를 알려 주고, "E KR 지불센터 [H(OI)]"를 전송함으로써 해당하는 OI에 대한 지불을 지불센터가 수행했다는 증거로 삼는다. 만약 이전 단계에서 문제가 발생했다면 이 "배달 요구 메시지"가 전송되지 않으므로, ⑦-1 메시지를 수신한 쇼핑서버는 지불이 수행되었다는 것을 확인할 수 있다. 거래ID에는 고객의 ID와 함께 그 고객의 몇 번째 거래인가를 나타내는 일련번호가 있으므로, 이 메시지에 대한 재전송 공격은 불가능하다. "E KU 쇼핑서버 [거래ID]" 부분도 암호화의 대상이 되는 메시지의 길이가 짧기 때문에 관용 암호 알고리즘을 사용하지 않고, 공개키 암호 알고리즘만을 사용하여 암호화하였다.

□ **배달 요구 응답 메시지(⑦-2)**

E KR 쇼핑서버 **[H(H(OI))]**

⑦-1 메시지를 수신한 쇼핑서버는 수신 확인을 알려주기 위해 ⑦-1 메시지에 포함되어 있는 내용 중, "H(OI)" 부분을 다시 해쉬함수의 입력으로 주어 그 결과에 자신의 전자서명을 생성하고, 그 전자서명 결과를 지불센터에게 전송한다. 이 메시지는 지불센터의 배달 요구 메시지를 쇼핑서버가 올바르게 수신했다는 의미이므로, 후에 쇼핑서버가 지불센터로부터 배달 명령을 수신한 사실이 없다고 주장할 때, 이를 반박할 수 있는 증거로 사용할 수 있다.

SafePay 프로토콜은 쇼핑서버가 인터넷에서 WWW 서버를 이용하여 서비스를 제공할 때, 지불 기능을 수행해 준다. 전자 상거래의 지불처리에서 발생할 수 있는 분쟁을 방지하고, 분쟁 발생시에 이를 해결해 줄 수 있는 기능에 중점을 두어 설계되었다. SafePay 프로토콜을 지원해 주는 전자지갑의 구현시에는 실세계에서의 사용 환경과 가장 유사한 환경을 제공해 줄 수 있도록 하였다. 다음에는 SafePay 프로토콜과 이를 지원해 주는 전자지갑의 구현 과정에 대해서 알아 본다.

4. SafePay의 구현

4.1 구현의 일반적인 사항

SafePay의 구현은 크게 전자지갑, 쇼핑서버, 지불센터, 인증서버의 구현으로 구분할 수 있다. 각 구성 요소의 구현을 위해 사용한 운영체제와 컴파일러는 <표 2>와 같다.

〈표 2〉 SafePay의 구현에 사용된 운영체제와 컴파일러

	운영체제	컴파일러
전자지갑	Windows 98	Borland C++ Builder
쇼핑서버	Linux Redhat 5.1	Gcc
지불센터	Linux Redhat 5.1	Gcc
인증서버	Linux Slackware 3.4	Gcc

<표 2>외에도 쇼핑서버에서 운영되는 WWW 서버는 Apache 서버 1.3.0을 사용하였고, 브라우저의 플러그-인 프로그램은 Visual C++ 4.2를 사용하였다.

SafePay의 구현을 위해 사용한 암호 모듈은 SSLeay이다. SSLeay 0.8.1과 SSLeay 0.9.0을 모두 사용할 수 있다. SSLeay에서 제공해 주는 함수를 그대로 구현에 사용하면 코드가 복잡해지므로 전자지갑, 쇼핑서버, 지불센터에서 사용하기 편리하도록 몇몇 함수들을 모아 하나의 모듈 단위로 재편집하였다. SSLeay에서 제공하는 함수 자체가 변경된 것은 아니고, 각 함수들의 구성을 다르게 했을 뿐이다.

□ 상품 주문과 상품정보 전송

사용자가 쇼핑서버에서 특정 물품을 구입하려면 상품을 전시하는 화면에서 "주문" 버튼을 클릭한다. 이 버튼을 클릭하면 연결되어 있는 CGI 프로그램이 구동된다. 이 CGI 프로그램은 주문된 상품의 정보를 정리하여 하나의 스트링으로 표현한 다음, 이를 사용자의 Netscape 브라우저로 전송하는 역할을 담당한다. Netscape 브라우저로 전송된 스트링은 전자지갑을 구동시키고 자신은 전자지갑의 장바구니에 저장되어야 한다.

□ 플러그-인 프로그램과 전자지갑의 구동

쇼핑서버에서 상품의 주문 버튼을 클릭하면 쇼핑서버의 CGI 프로그램에서 상품명, 가격, 상품코드, 개수로 구성되는 정보를 화일에 저장하여 embed tag를 이용하여 브라우저로 전송한다. 플러그-인 프로그램은 이 정보를 하드 디스크에 저장하고 전자지갑을 구동한다[11]. 플러그-인 프로그램의 역할은 쇼핑서버에서 전송되는 정보를 수신하여 하드 디스크의 특정 부분, 즉 전자지갑이 참조하는 부분에 저장하고, 전자지갑을 구동시키는 역할을 수행한다.

□ 플러그-인 구동을 위한 CGI 프로그램

사용자가 쇼핑서버에 접속하여 상품의 구입을 위해 "주문" 버튼을 클릭하는 데, 이 페이지의 소스 코드에는 (그림 6)과 같은 부분이 존재한다.

```
<form action = "http://seeme.chungnam.ac.kr/~totoro/
               cgi-bin/shop/shopping.cgi" method= "post">
<input type = "HIDDEN" name = "name" value = "Flower
               basket F">
<input type = "HIDDEN" name = "code" value = "10201">
<input type = "HIDDEN" name = "price" value = "50000">
<input type = "text" size = "6" name = "num" value = "1"
               MAXLENGTH = 3>
```

(그림 6) 쇼핑 페이지 소스 코드의 일부

이 코드의 의미를 보면 상품정보를 나타내는 네 가지의 정보(name, code, price, num)를 "http://seeme.chungnam.ac.kr/~totoro/cgi-bin/shop/shopping.cgi" 프로그램으로 전송한다. 이 CGI 프로그램은 사용자의 브라우저로부터 전송된 네 개의 정보를 이용하여 HTML 화일을 생성한다. 이 때 생성되는 HTML문서는 사용자 브라우저에서 플러그 인 프로그램을 구동하는 역할을 수행한다[11][12].

4.2 전자지갑의 구현

전자지갑의 구현에 관한 설명은 좀 더 세분화된 단계에 따라 진행되는 데, 다음과 같은 부분으로 세분화된다.

- ○ 사용자 ID 등록
- ○ 공개키/비밀키 쌍 관리
- ○ 장바구니 관리
- ○ 지불수단 관리
- ○ 거래내역 관리

각각에 대한 자세한 설명은 다음과 같다.

□ 사용자 ID 등록

전자지갑의 사용을 위해서 사용자는 ID와 이에 해당하는 패스워드를 전자지갑에 등록시켜 놓아야 한다. ID는 쇼핑서버에서도 저장/관리하기 때문에 사용자는 자신이 사용하려고 하는 ID를 쇼핑서버로 전송하여 사용 가능 여부를 확인한다. 쇼핑서버에서 다른 사용자가 동일한 ID를 사용하면 문제가 발생하기 때문이다. 사용자가 입력한 ID는 쇼핑서버로 전송되어 중복 여부를 검사한다. 중복되지 않는 ID라면 전자지갑에 패스워드와 함께 "passwd.ttt"화일에 저장된다.

패스워드는 해쉬함수에 입력되어 계산된 결과가 화일에 저장된다. 패스워드는 사용자의 인증과정에서도 사용되지만, 뒤에서 설명될 지불수단을 암호화할 때에도 사용된다. 그러므로, 사용자 인증에 사용되는 패스

워드는 해쉬함수를 두번 계산하여 그 결과를 저장하고, 한번 계산한 결과는 지불수단의 암호화에 사용된다.

□ 공개키/비밀키 쌍 관리

사용자ID와 패스워드를 등록하게 되면 사용자 인증 과정을 수행한 후에 전자지갑을 사용할 수 있다. 이 시점에서 전자지갑은 사용자의 공개키/비밀키 쌍이 존재하지 않는 것을 확인하고, 이 사실을 사용자에게 알려 준다. 이 시점에서 사용자가 공개키/비밀키 쌍을 생성할 수도 있고 연기할 수도 있다. 공개키/비밀키 쌍을 생성하면 공개키는 인증서버로 전송되어 인증서버가 발행해 주는 인증서를 받아 온다. 이 단계에서도 전자지갑과 인증서버 사이에 설치되어 있는 소켓을 이용하여 통신한다. 인증서는 공개키와 사용자의 ID로 구성된 스트링에 대한 인증서버의 서명으로 구성된다. 비밀키는 생성 과정에서 사용자가 입력한 패스프레이즈를 암호키로 사용하여 암호화한 후에 저장된다.

□ 장바구니 관리

SafePay에서는 사용자가 주문하려고 하는 상품의 정보를 전자지갑에서 관리한다. 상품정보는 앞에서 설명한 플러그-인 기법에 의해 하드 디스크의 특정 부분에 계속 갱신되며 저장된다. 전자지갑에서는 상품정보가 저장되는 화일을 주기적으로 검사하여 변경된 내용을 전자지갑의 장바구니 부분에 반영해 주는 방법을 이용한다. 이미 구동되어 있는 전자지갑이 있는 경우에 새로운 상품을 주문하면 이 상품정보는 현재 구동되어 있는 지갑의 장바구니에 등록되어야 한다. 그러나 플러그-인 프로그램은 이러한 상황을 고려하지 않고, 단순히 전자지갑 프로그램을 실행하는 동작을 수행하기 때문에 주문시마다 새로운 전자지갑을 구동시킨다[11]. 이러한 경우에는 전자지갑이 구동되면서 이미 구동되어 있는 전자지갑이 있는 지를 검사하여, 있으면 상품정보를 기존의 전자지갑이 참조하는 화일에 추가하고 자신은 종료되는 방법을 사용해야 한다. 다음은 이미 구동되어 있는 전자지갑이 있는 지를 검사하는 함수이다.

```
MutexHandle = CreateMutex(NULL, true, "xius");
```

□ 지불수단 관리

SafePay에서의 지불수단은 신용카드이다. 그러므로, 신용카드 정보 즉, 신용카드 회사, 회원 번호, 거래 금융 기관, 만기일, 비밀 번호 등을 미리 전자지갑에 등록한다. 미리 신용카드 정보를 등록해 놓음으로써 주문을 할 때마다 신용카드 정보를 입력해야 하는 번거로움을 피할 수 있다. 그러나, 이 신용카드 정보들을 평문으로 사용자의 컴퓨터에 저장해 놓으면 노출의 위험성이 크기 때문에 이 정보들을 암호화하여 저장한다. 암호화에 사용하는 키는 사용자가 사용자 인증을 위해 입력했던 ID와 패스워드중에서 패스워드를 이용한다. 패스워드는 앞에서 설명했던 바와 같이 사용자의 인증과 지불수단의 암호화에 사용되기 때문에 각각의 경우에 해쉬함수 계산의 횟수를 다르게 한다[13].

□ 거래내역 관리

정상적으로 종료된 거래에 대해서 SafePay는 사용자의 전자지갑에 지불센터의 전자서명이 포함되어 있는 문서를 저장해 놓는다. 이 정보가 영수증의 역할을 수행한다. 영수증은 거래에 사용됐던 상품정보와 지불정보의 해쉬함수 결과에 지불센터가 전자서명을 계산해 놓은 형태로 구성되어 있다. 지불센터가 영수증을 생성하여 전자지갑으로 전송하면 전자지갑은 이의 내용을 확인하여 올바르게 구성된 영수증일 경우에만 전자지갑의 거래내역 부분에 저장한다. 만약, 문제점이 발견되면 전자지갑은 지불센터에게 이를 알려 주고 그 시점까지 진행됐던 모든 동작을 취소한다.

4.3. 쇼핑서버의 구현

쇼핑서버에서 수행하는 작업은 다음과 같이 크게 분류할 수 있다.

- 사용자ID 관리
- 배달 업무

각 작업에 대한 설명은 다음과 같다.

□ 사용자 ID 관리

사용자 ID 관리는 앞에서 이미 설명된 바와 같이 전자지갑의 사용자 ID가 거래 ID의 일부로 사용되기 때문에 여러 사용자들끼리의 ID 충돌을 방지하기 위한 것이다. 전자지갑 사용자의 ID는 쇼핑서버에서도 저장한다. 등록된 사용자들의 ID를 특정한 화일에 모두 저장하고 새로운 사용자가 ID를 등록할 때에 이미 등록되어 있는 ID인지를 검사한다. 사용자 관리의 편리를 위해 쇼핑서버는 사용자 한 명당 하나의 디렉토리를 할당한다. 이 디렉토리에 각 사용자와 통신한 OI 정보, 이에 대한 전자서명 등을 저장한다. 디렉토리의 이름은 각 사용자의 ID를 사용한다.

배달 업무

쇼핑서버가 수행하는 또 하나의 작업은 주문된 상품의 배달이다. 배달방법 자체는 SafePay에서 다루는 부분이 아니다. 지불센터가 전송한 메시지를 참조하여 그에 해당하는 OI 정보를 추출하여 배달할 상품을 선택하는 과정을 제공한다. 쇼핑서버는 사용자와 통신한 OI 정보를 각 사용자에게 할당된 디렉토리에 저장한다. 이때 디렉토리의 구분은 사용자의 ID로 결정한다. 저장된 정보와 지불센터가 전송하는 배달명령에도 거래 ID가 포함되어 있다. 거래 ID는 유일한 정보이므로 다른 거래 ID와 충돌이 발생할 수 없다. 지불센터가 전송한 정보 중에서 거래 ID를 추출해 같은 거래 ID를 포함하는 OI 정보를 검색하여 배달에 이용한다.

4.4 SafePay의 안전성

SafePay는 암호기술의 사용으로 앞에서 지적되었던 전자지불 시스템의 문제점들을 모두 해결하였다. 이에 관련한 설명을 기존의 전자지불 시스템과 비교하여 설명하면 다음과 같다.

신용카드 정보의 노출과 저장

신용카드 정보는 전자지갑의 특정한 파일에 저장된다. 사용자가 인증과정에서 입력한 패스워드를 해쉬함수로 계산한 결과값을 암호키로 사용하여 신용카드 정보를 암호화하여 저장한다. 지불 메시지에 포함되는 신용카드 정보의 내용을 쇼핑서버는 볼 수 없고 지불센터에서만 확인할 수 있으므로 쇼핑서버에 의한 신용카드 사용도 막을 수 있다. 이러한 기능은 대부분의 지불 시스템에서는 제공해 주지 않는 것이다.

지불처리의 주체

SafePay에서의 지불처리는 사용자의 요구에 의해서 가능하다. 사용자가 지불을 요구하기 위해서 필요한 정보들은 다음과 같다.

- 사용자 인증과정에 필요한 패스워드
- 사용자 비밀키의 복호화를 위한 패스프레이즈
- 신용카드의 비밀번호

앞의 세가지 정보에서 올바른 값을 입력하지 못하는 것이 하나라도 있으면 지불처리를 요구할 수 없다. 이는 사용자 자신 이외에 지불처리를 요구할 수 있는 사용자는 없다는 것을 의미한다. 쇼핑서버가 신용카드 정보를 알고 있더라도 앞의 정보를 모르면 지불처리를 요구할 수 없으므로 지불주체가 사용자 자신일 수 밖에 없다는 것을 알 수 있다. 그리고 (그림 3)에서 알 수 있듯이 지불이 수행될 때, 사용자가 직접 지불센터와 통신하기 때문에 쇼핑서버가 중간에서 속임수를 사용할 수 있는 기회를 제공하지 않는다. 쇼핑서버가 지불센터와 통신하는 경우는 배달을 명령하는 단계로써 이 시기에는 이미 지불센터가 거래의 근거자료로 사용할 수 있는 모든 정보를 확보한 이후이기 때문에 쇼핑서버가 속임수를 사용할 수 없다. SET에서는 쇼핑서버가 사용자와 지불센터 사이에 위치하기 때문에 통신되는 메시지들에 대해 속임수를 사용할 수 있는 기회를 제공한다.

영수증 부재

거래에 사용됐던 상품정보와 지불정보의 해쉬함수 결과에 대해 지불센터가 전자서명을 계산한 것을 영수증으로 사용하고 있다. 상품정보와 지불정보의 해쉬함수 결과는 사용자와 쇼핑서버가 모두 저장하고 있는 정보이기 때문에 지불센터의 전자서명 내용을 확인할 수 있다. 또한, 사용자가 영수증의 내용에 문제가 있다고 판단하면 거래 전체를 취소할 수 있으므로 영수증의 문제가 더 이상 확대되지 않는다. 대부분의 지불 시스템에서는 영수증을 제공하지 않으므로 영수증을 제공하는 기능은 타 시스템과 비교해 큰 장점으로 작용한다. SET에서도 영수증은 제공하고 있다. 그러나, SafePay에서는 사용자가 영수증의 내용을 확인한 후, 내용에 이상이 있다고 판단되면 지불처리가 완료되기 이전에 거래 전체를 취소할 수 있다. SET에서도 거래 취소와 관련한 기능을 제공하지만 SafePay처럼 취소과정이 지불과정 내부에 포함되어 있지 않고 취소 부분을 별도로 구현하여 제공해야 한다.

장바구니 기능의 역효과

장바구니는 사용자의 컴퓨터에 위치한다. 그러나, 사용자가 쇼핑서버로 전송하는 내용에는 장바구니에 저장되어 있는 값으로 구성된 상품정보에 사용자의 전자서명이 포함되기 때문에 사용자에 의한 내용 변조가 불가능하다. 사용자는 쇼핑서버와 지불센터에 동일한 상품정보를 전송해야 하므로 가격 등의 정보를 변경하여 지불처리를 요구할 수 없다. 상품과 관련한 정보를 사용자의 컴퓨터에 보관하는 것은 실제 쇼핑에서와 동일한 환경을 제공해 준다.

□ SSL의 취약성

암호모듈을 교체하면 다양한 암호 알고리즘을 수용할 수 있다. 현재 구현된 SafePay 시스템에서는 128비트의 키를 사용하는 IDEA를 관용 암호방식으로 사용하고 있다. 이는 40비트의 키를 사용하는 SSL에 비해 월등한 보안 강도를 유지시켜 준다.

□ 간단한 인증체계

SET이 실제 전자지불에 상용화되지 못하고 있는 이유들 중에 하나가 인증체계의 복잡성이다. SafePay는 시스템 구성원들 모두가 신뢰하는 인증기관을 설치하는 모델을 사용함으로 인증 체계의 복잡성을 단순화시켰고, 실제 구현과 사용에 적합하도록 하였다.

5. 향후 연구 방향

현재 운영되는 쇼핑서버들이 제공하는 전자지불 방법들은 낮은 강도의 암호 알고리즘 사용, 신용카드 정보의 노출, 영수증의 부재 등 많은 보안상의 약점들을 가지고 있다. 이러한 상황에서는 전자상거래의 지속적인 발전을 기대할 수 없으며, 금융사고 등의 부작용을 발생시킬 수 있다.

본 논문에서 제안하고 구현한 SafePay는 WWW 서버를 기반으로 운영되는 쇼핑서버와 연계하여 사용할 수 있고, 전자지갑은 사용자마다 자신의 컴퓨터에 설치하여 사용한다. 현재 운영되고 있는 쇼핑서버들이 내재하고 있는 지불 프로토콜 관련 문제점을 모두 해결하였으며, 전자지갑 자체에서 제공하는 보안기능을 통해 더욱 안전한 쇼핑 환경을 제공한다. 전자지갑은 사용자가 실제 쇼핑을 할 때와 가장 유사한 환경을 제공하는 데에 주력하였다. 또한, 앞으로 전자 상거래가 활성화될 것에 대비해 신용카드의 비밀번호 입력 등 현재 요구하지 않고 있는 보안 기능을 추가하였다.

그러나, 플러그-인 기술을 적용하였기 때문에 사용가능한 브라우저가 Netscape으로 제한되는 문제점을 가지고 있다. 또한 현재 구현한 SafePay 시스템은 전자지갑을 1인용으로 설계하였기 때문에 전자지갑을 하나만 설치하여 여러 사용자가 안전하게 사용할 수 있는 시스템의 개발이 필요하다.

좀 더 안정적이고 견고한 공개키 기반 구조가 확립되어야 한다는 점도 앞으로 수행되어야 할 분야이다. 현재 구현된 시스템에서는 공개키의 스트링에 사용자 ID를 연결하여 인증서버의 전자서명을 붙인 형태의 인증서를 사용하고 있다. 그러므로, 안전하면서 응용 시스템에 종속되지 않는 공개키 관리 기반의 구축이 시급하다.

참 고 문 헌

[1] 남경두외 1인, "인터넷 무역 길라잡이", 정보문화사, 1998

[2] 한국정보통신진흥협회, 정보화사회, 1998년 3,4월호

[3] 김춘길, "인터넷 전자상거래", KRNET '98 특강자료집, 1998

[4] Terry Bernstein외 3인, "Internet security for business", John Wiley & Sons, Inc., 1996

[5] 도경구, "전자화폐의 유통 시스템 모델," KRNET '98 발표자료집

[6] Master card & VISA, "SET : Secure Electronic Transaction Specification Version 1.0," 1997

[7] Lincoln D. Stein, "Web security," Addison-Wesley, 1998

[8] Derek Atkins, "Internet security," New riders, 1996

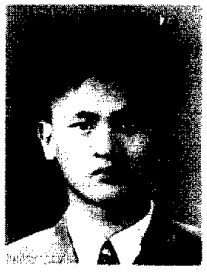
[9] EFF DES Cracker Project, http://www.eff.org/descracker

[10] Bruce Schneier, "Applied cryptography," John Wiley & Sons, Inc., 1994

[11] Zan Oliphant, "Programming Netscape plug-ins," Sams net, 1996

[12] Shishir Gundavaram, "CGI Programming on the World Wide Web," O'Reilly & Association, Inc., 1996

[13] William Stallings, "Network and internet security," Prentice Hall, 1995



박 현 동

e-mail : hdpark@esperosun.chungnam.ac.kr

1995년 충남대학교 전산학과(학사)

1997년 충남대학교 대학원 컴퓨터과학과(석사)

1997년~현재 충남대학교 대학원 컴퓨터과학과 박사과정

관심분야 : 네트워크 보안, 암호학, 전자상거래 보안

이 은 성

e-mail : eslee@esperosun.chungnam.ac.kr
1998년 충남대학교 컴퓨터과학과 대학원 입학(석사과정),
관심분야 : 네트워크 보안

송 상 헌

e-mail : shsong@esperosun.chungnam.ac.kr
1996년 배재대학교 전자계산학과 졸업(학사)
1998년 충남대학교 대학원 컴퓨터과학과(석사)
1998년~현재 충남대학교 대학원 컴퓨터과학과 박사과정
관심분야 : 보안 프로토콜, PKI, 전자지불시스템

강 신 각

e-mail : sgkang@pec.etri.re.kr
1984년 충남대학교 전자공학과 (학사)
1987년 충남대학교 전자공학과 (석사)
1984년~현재 한국전자통신연구원 표준연구센터 표준기획연구팀 책임연구원
관심분야 : 통신망 보안, 멀티미디어 통신.

박 정 수

e-mail : jspark@etri.re.kr
1992년 경북대학교 전자공학과 (학사)
1994년 경북대학교 전자공학과 (석사)
1994년~현재 한국전자통신연구원 표준연구센터 연동표준연구팀 연구원
관심분야 : 인터넷 보안, 멀티미디어 보안

류 재 철

e-mail : jcryou@esperosun.chungnam.ac.kr
1985년 한양대학교 산업공학과 (학사)
1988년 Iowa State University 전산학과(석사)
1990년 Northwestern University 전산학과(박사)
1991년~현재 충남대학교 컴퓨터과학과 부교수
관심분야 : 컴퓨터통신보안, 전자상거래, 분산처리