經營科學
第18卷 第1號
2001年 5月

소액 지불 전자상거래 환경을 위한 프로토콜의 설계 및 구현

손병록* · 박기현** · 유상진***

Design and Implementation of a Micropayment Protocol in Electronic Commerce Environments

Byung-Rok Son* · Kee-Hyun Park** · Sang-Jin Yoo***

Abstract

An electronic micropayment system, one of electronic payment systems, is suitable especially when a small amount of money is to be paid frequently in order to purchase on-line goods (i.e., database search, software distribution, electronic news service, etc). In addition, since the amount of payment is small, possible damages caused by system failures are lower than other payment systems.

In this paper, a micropayment protocol in electronic commerce environments on the Internet is proposed, based on the PayWord system. And a micropayment electronic commerce system which executes the protocol proposed is implemented. Unlike the PayWord system, however, the micropayment protocol proposed in this paper is designed in such a way that a merchant does not need to request a payment at the end of every business day since a customer cannot purchase goods beyond length of hash chains, which is specified on a certificate. In addition, the system is able to check the validity as well as the duplicate spending of hash chains. The electronic micropayment system with the proposed protocol is consists of Customers, Merchants, and Brokers. Customers are implemented on Windows NT 4.0 using VC++. Merchants are implemented on Solaris 2.5.1 and gcc 2.8.0 using Netscape Web Server and CGI methods. HP UX 10.20 is used for Brokers.

* 계명대학교 대학원 컴퓨터공학 전공
** 계명대학교 공학부 교수
*** 계명대학교 경영학부 교수

1. 서 론

인터넷의 상업적인 이용은 실제 구매에까지 이루어지게 되었으며, 광고, 생활용품, 소프트웨어, 온라인 정보 등과 같은 상품들이 전시되고 판매되는 사이버 쇼핑몰이 등장하게 되었다. 우리 나라에서도 현대백화점[4], 메타랜드[1] 등 여러 쇼핑몰이 개장되어 거래가 이루어지고 있다. 이처럼 컴퓨터 통신망을 통해 기업, 개인, 정부 등의 주체가 상품, 서비스 등의 객체를 상호간 판매, 구매, 광고하는 전반적인 거래 행위를 전자상거래(EC : Electronic Commerce)라고 한다[2, 3, 6, 12]. 전자상거래는 신속성, 편의성 및 광역성으로 인하여 매우 빠른 속도로 발전하고 있으며, 이에 대한 기업체들의 수요도 해마다 급격하게 늘고 있는 추세이다. 전자상거래의 주요 요건들 중의 하나는 안전성과 효율성을 갖춘 전자 지불 시스템을 개발하는 것이다. 전자 지불 시스템은 거래에 참여하는 고객, 상인 등이 서비스 및 그것에 대한 대가를 안전하고 효과적으로 주고받을 수 있는 정보 전달 및 대금 지불 체계를 말한다.

이 중에서 소액 지불 시스템은 온라인 상품인 데이터베이스의 검색, 소프트웨어 판매, 전자신문 및 전자저널 서비스 등과 같은 상품 구매에 대해서 소액의 금액을 빈번히 지불해야 할 경우에 적합한 시스템이다. 또한 지불 금액이 소규모이므로, 시스템의 오류로 인한 피해가 다른 지불시스템들 보다도 적은 장점이 있다.

본 연구에서는 인터넷으로 상품을 구매하는데 적합한 소액 지불 프로토콜(protocol)을 제안하고, 이를 실행하는 전자상거래 시스템을 설계 구축한다. 본 연구의 소액 지불 시스템은 PayWord[16]를 기반으로 하고 있으나, 상인이 중개인에게 영업일마다 매번 정산 요청을 할 필요가 없도록 설계되었으며, 체인의 유효성 및 이중 지출 등을 점검할 수 있다. 소액 지불 시스템은 인터넷상에서 Client-Server 방식으로 구현된다. 본 논문의 구성은 다음과 같다. 2절에서는 지금까지 연구된 전자 지불 시스템들을 살펴본다. 3절 및 4절에서는 본 연구에서 제안하는 소액 지불 시스템의 설계 및 구현을 각각 설명한다. 마지막으로 5절에서는 결론 및 향후 연구과제 등을 다룬다.

2. 관련 연구 동향

2.1 지불 브로커(Payment Broker) 시스템

기존의 전통적인 지불 방식(신용카드, 은행계좌 이체, 직불카드 등)을 인터넷의 지불 시스템으로 안전하고 편리하게 연결해주는 시스템인데, 나중에 설명할 다른 시스템들과는 달리, 독자적인 지불 방식은 지니고 있지 않다. 대표적인 예로서는 전자상거래 환경에서 신용카드 번호나 은행계좌 번호를 안전하게 주고받을 수 있도록 중계해 주는 시스템을 들 수 있다.

이 방식의 장점은 자신의 대금처리 방식, 즉 금융 환경을 필요로 하지 않기 때문에 지불 정보(신용카드 번호, 계좌번호와 비밀번호 등)를 전달하는 트랜잭션만 안전하고 편리하게 처리해 줌으로써, 비교적 쉽게 구성할 수 있다는 것이다. 그러나 지불 브로커 시스템은 자기 자신의 시스템을 운영하고 처리하는 비용뿐만 아니라, 다른 지불 방식들도 함께 수용해야 하는 오버헤드(신용카드 조회 비용, 신용카드 수수료, 계좌이체 수수료 등)로 인한 이중 부담이 있어서 운용 비용이 상대적으로 비싸다. 이런 지불 브로커 시스템을 위하여 개발된 프로토콜들로서는 FV(First Verture)[10]와 CyberCash [7], SET(Secure Electronic Transaction)[13] 등을 들 수 있다.

2.2 전자화폐(Electronic Cash)시스템

전자화폐 시스템은 외부의 다른 지불방식을 직접적으로 이용하지 않고, 시스템 내부에서 자신만의 신용을 기반으로 화폐와 같은 부채나 가치를 발행하는 형태이다. 전자화폐 시스템의 장점은 외부

지불 방식의 직접적인 도움이 필요 없기 때문에, 시스템 흐름이 간단하며 지불 브로커 시스템보다 발전적인 상거래 방식으로 간주된다.

그러나 이런 방식이 성공적으로 운용되기 위해서는, 전자화폐 발행의 법적인 효력, 기존 금융기관들과의 역할 중복성 등의 제도적인 문제가 먼저 해결되어야 한다. 또한 전자화폐의 신빙성 확인, 전자화폐의 이중 지출(double spending) 방지, 익명성(anonymity) 확보 등의 문제들을 충분히 고려해야 한다. 대표적인 전자화폐 시스템으로서는 ECash[8]가 있다.

2.3 소액 지불시스템

소액 지불시스템은 전자화폐 시스템의 특수한 형태로서, 소량의 금액 지불을 전문적으로 처리하기 위하여 고안되었다. 특히 인터넷상의 전자상거래 방식 가운데, 온라인 상품인 데이터베이스 검색, 소프트웨어 판매, 전자뉴스 제공 등과 같은 서비스 등에는 소액 지불만으로도 충분하다. 소액 지불 시스템은 처리비용을 최소로 하여 지불 처리에 소요되는 비용을 상품의 가격보다 낮춘 지불 시스템이다. 지불 금액이 소규모이므로, 시스템의 오류로 인한 피해가 다른 지불 시스템들보다도 적은 장점이 있다. 전자지불 시스템의 대표적인 프로토콜들로서는 MPTP[15], Millicent[14], PayWord[16] 등이 있다.

2.3.1 Millicent

Millicent는 고객, 중개인, 상인 등의 세 부분으로 구성되며, 스크립(Scrip)이라는 전화화폐를 기반으로 하는 시스템이다. 스크립은 금액을 포함하고 있으며, 거래가 이루어질 때 지불의 수단으로 사용된다.

Millicent의 처리과정을 살펴보면, 중개인은 각 상인으로부터 스크립 판매 라이센스를 받아와서 고객의 요청에 따라 스크립을 판매한 후, 판매된 스크립 만큼의 돈을 상인에게 지불한다. 만약 이전에 거래가 없었던 새로운 상인과 고객이 거래를 하려고 할 경우에는, 먼저 고객이 중개인에게 실제의 돈을 주고 스크립을 구매한 후, 상인에게 상품을 요청할 때 이 스크립을 함께 보낸다. 상인은 스크립의 유효성을 검사하고 상품과 거스름돈(상품의 가격을 감한 스크립)을 고객에게 넘겨줌으로써 거래가 완료된다.

Millicent는 상인마다 고유의 화폐를 사용함으로써 독립적으로 화폐의 이중 지출, 위조, 변조를 막을 수 있다는 장점이 있다. 반면 고객이 새로운 상인과 거래를 원할 경우, 중개인으로부터 그 상인의 스크립을 구입해야 하기 때문에 이에 대한 지연시간이 발생한다. 또한 고객이 스크립을 구입할 때 실제 지불이 일어나기 때문에 스크립의 데이터가 손실되면 실제 돈을 잃어버리는 것이 된다.

2.3.2 PayWord

PayWord 역시 고객, 중개인, 상인 등의 세 부분으로 구성된다. 고객은 중개인에게 계좌를 설립하고 중개인은 중개인의 이름(B), 사용자의 이름(U), IP 주소(A_U), 고객의 공개키(PK_U), 유효기간(E) 그리고 다른 기타 정보(I_U)를 포함하는 전자서명된 PayWord 인증서를 발급한다. 인증서 C_U의 형식은 다음과 같다 :

$$C_U = \{B, U, A_U, PK_U, E, I_U\}SK_B$$

인증서는 사용자가 PayWord 해쉬 체인(hash chain)을 만들 수 있는 권한을 부여한다. 또한 인증서는 일정기간이 지나면 중개인이 고객의 계좌가 좋은 상태인지를 검사하여 재발급한다. 고객은 PayWord 체인을 다음과 같이 만든다 : 마지막 PayWord인 w_n는 임의로 정하고, w_n을 제외한 나머지 PayWord들은 $w_i = h(w_{i+1})$ (i = n-1, n-2, …, 0)을 계산함으로써 체인을 생성한다. 그런 다음, 고객은 그 체인에 대해 아래와 같은 Commitment를 만든다.

$$M=\{V, C_U, w_0, D, I_M\}SK_U$$

(V : 상인이름, D : 현재시간, I_M : 기타 정보)

PayWord는 상인마다 유일한 체인을 사용한다. 고객은 지불할 상인마다 별도의 Commitment를 만들어야 하며, Commitment에서 기술된 V 이외의 상인에게는 이 체인이 전혀 의미가 없다. PayWord의 지불과정은 다음과 같다 : 고객의 i번째 지불은 (w_i, i)로 구성되며 상인은 w_{i-1}을 사용한 해쉬 연산으로 유효성을 확인할 수 있다. 그 날의 마지막에 상인은 각 고객에게 받은 마지막 지불인 $P_l=(w_l, l)$과 그에 대응하는 Commitment를 함께 중개인에게 보낸다. 중개인은 l번의 해쉬 함수를 반복 수행하여 w_l을 확인한 후, 고객의 계좌에서 l 만큼의 금액을 청구하여 상인의 계좌로 지급한다.

PayWord는 각 상인에 대한 고유한 PayWord를 사용하기 때문에 상인이 독립적으로 PayWord의 이중 지출, 위조, 변조를 검사할 수 있다는 장점이 있다. 반면 새로운 고객과 거래를 할 때마다 새로운 PayWord 체인을 만들어서 상인에게 Commitment를 전달하여야 한다. 또한 고객이 자신의 신용한도액을 초과하는 PayWord 체인을 만들어서 정산하기 전에 사용할 수 있는 문제점이 있다.

3. 소액 전자 지불 프로토콜의 설계

본 논문에서 제안하는 소액 지불 프로토콜은 PayWord 시스템을 기반으로 하여 설계하되, 다음과 같은 다른 점들을 고려한다 :

(1) 고객은 새로운 해쉬 체인을 형성할 때마다 중개인으로부터 체인 길이에 대한 정보가 포함된 인증서를 받는다. 기존의 PayWord 방식보다 인증서 발급 회수가 많아지는 단점이 있지만, 체인 길이 이상의 상품 구매를 사전에 방지할 수 있는 이점이 있다. 또한 인증서의 유효기간이 갱신될 때마다 새로운 인증서를 발급해야 하므로, 발급되는 인증서에 새로운 유효 기간을 가진 해쉬 체인을 포함시킨다면 실제로 인증서 발급 회수는 체인 형성 수만큼 늘어나지 않는다.

(2) 고객은 인증서에 명시된 체인 길이 만큼의 상품 구매만 가능하므로, 상인이 서둘러서 정산 요청을 할 필요가 없으며, 한꺼번에 청구할 수도 있다. 따라서, PayWord와는 달리, 상인은 영업일이 끝날 때마다 매번 중개인에게 정산을 요청할 필요가 없다.

3.1 해쉬 체인의 생성과 인증서 발급

해쉬 체인은 상인마다 고유하게 생성되고 사용되어 진다. 다음과 같은 환경이 조성되면, 새로운 해쉬 체인을 형성하기 위하여 고객이 요청한다.

(1) 거래하고자 상인과 이전에 한번도 거래가 없었을 경우,
(2) 해당 상인에 관한 해쉬 체인을 모두 소비하였을 경우, 그리고
(3) 해쉬 체인(혹은 인증서)의 유효기간이 지났을 경우 등이다.

고객은 중개인에게 지불한 금액에 해당하는 길이의 해쉬 체인을 받는다. 해쉬 체인 생성은 암호학적으로 안전하다고 알려져 있는 해쉬 함수인 MD5를 이용한다 [17]. 이때, 체인의 루트 값인 w_0는 실제 지불에는 사용하지 않고 체인의 유효성을 검사하기 위해서 사용된다. 해쉬 체인이 실제 지불에 사용되기 위해서는 체인에 대해서 중개인의 인증서가 필요하다. 고객은 본인의 고유번호 (u_{id}), 체인의 루트 값 (w_0), 체인의 길이 (w_{len})을 중개인에게 보내는데, 제삼자의 도청을 막기 위해서 중개인의 공개키 (B_{pk})로 암호화하여 보낸다.

중개인은 자신의 비밀키를 사용해서 복호화하고 고객의 계정을 조회하여 해쉬 체인의 금액이 신용

한도액을 초과하지 않는지 검사한다. 체인의 금액이 신용한도액을 초과하지 않는다면, 인증서 C를 작성하여 고객에게 보낸다.

$$C = \{c_{id},\ v_{id},\ u_{id},\ w_0,\ w_{len},\ e_t\}B_{sk}$$

여기서, c_{id}는 인증서 고유번호, v_{id}는 상인의 고유번호, , u_{id}는 고객의 고유번호, w_0는 체인의 루트, w_{len}는 체인의 길이, e_t는 인증서의 유효 사용기간, B_{sk}는 중개인의 비밀키를 나타낸다. 인증서는 중개인의 전자서명을 첨부하여 고객으로 보낸다. 이 인증서는 유효 사용기간 내에 고객이 지불한 체인 길이에 대해서 정산함을 보장한다. 지불 과정에서 상인은 체인 길이를 초과하는 지불을 인정하지 않음으로써, 고객은 자신의 신용한도액을 초과하는 거래를 할 수 없게 된다.

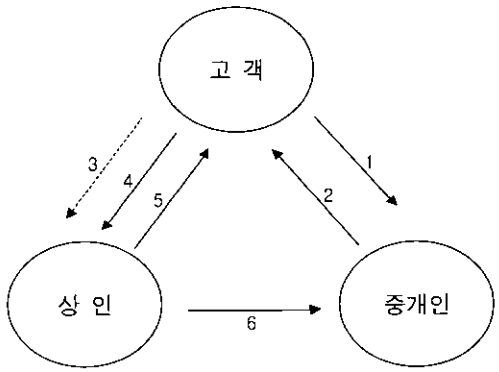
3.2 지불 서비스

고객은 인증서를 상인에게 보낸다. 상인은 중개인의 전자서명을 확인함으로써 유효기간, 체인의 루트, 체인의 길이 등, 앞으로 고객과 거래에 필요한 정보를 확인할 수 있다. 인증서는 처음으로 거래를 하는 상인에게만 보낸다. 지불은 체인과 체인 인덱스 (w_p, p)를 고객이 상인에게 보냄으로써 이루어진다.

예를 들면, 체인의 요소하나의 가치를 1원이라 하고 지금까지 체인이 i번째까지 사용되었다. 고객이 n원 짜리 상품을 구매했을 때, 지불은 지불할 체인 $w_{i+1}, w_{i+2}, \cdots, w_{i+n}$ 중에서 마지막 체인과 체인 인덱스 $(w_{i+n}, i+n)$으로 구성된다. 상인은 w_{i+n}을 $i+n$ 만큼의 해쉬 함수를 실행한 후, w_i와 비교하여 체인의 유효성을 검사한다. 그리고 $(i+n)-i$를 통해서 n 만큼의 지불을 확인할 수 있다. 지불한 인덱스의 값이 체인의 길이를 넘어선다면, 체인의 유효 길이를 넘어선 신용한도액의 초과로 보고 이 지불을 인정하지 않는다. 해쉬 체인은 그 상인에게만 유효하므로, 상인은 독자적으로 고객의 체인에 대한 이중 지출을 검출할 수 있다. 즉, 현재 지불한 체인의 인덱스가 직전에 지불한 인덱스보다 작거나 같다면 이중 지출로 판단한다.

3.3 정 산

상인은 그 날까지 받은 고객의 가장 최근 지불과 체인 인덱스 (w_l, l)을 중개인의 인증서와 함께 중개인에게 보냄으로써 정산을 요구한다. 상인은 고객으로부터 중개인의 인증서를 받는데, 그 인증서에는 고객이 사용할 수 있는 체인의 길이가 포함되어 있어서 고객이 지불할 때마다 상인이 신용한도 초과 유무를 검사할 수 있다. 따라서 PayWord와는 달리, 상인은 매일 중개인에게 정산을 요구할 필요는 없다. 중개인은 해쉬 함수를 수행하여 체인의 유효성을 확인한다. 그리고 고객의 체인이 몇 번째까지 정산되었는지를 확인한 후, 현재 정산을 요구한 체인의 인덱스와 가장 최근에 정산된 인덱스의 차이만큼의 금액을 고객의 계좌에서 인출하여 상인에게 지불한다. 본 연구에서 제안한 소액 지불 프로토콜에서의 자료 흐름은 <그림 1>과 같다.



1. 계좌 설립 및 인증서 요청 : $\{u_{id},\ w_0,\ w_{len}\}B_{pk}$
2. 인증서 (C) 수신 : $\{c_{id},\ v_{id},\ u_{id},\ w_0,\ w_{len},\ e_t\}B_{sk}$
3. 인증서 전달(첫번째 거래) : C
4. 지 불 : (w_p, p)
5. 상품 배달
6. 정산 요청 : $\{C, w_l, l\}$

〈그림 1〉 소액 지불 프로토콜의 자료 흐름

4. 소액 전자 지불 시스템 구현

4.1 시스템 구성

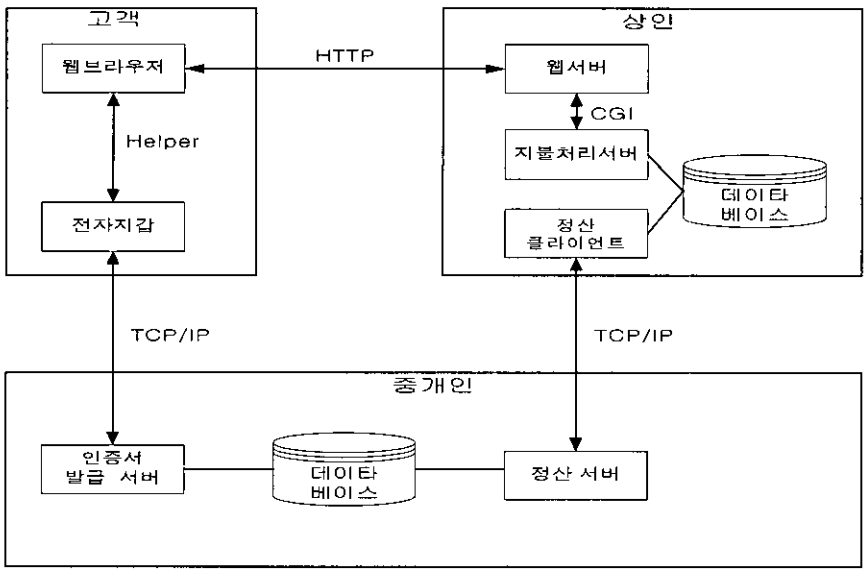
본 연구가 제안하는 소액 지불 프로토콜을 기반으로 한 전자상거래 시스템은 크게 고객, 상인, 중개인으로 구성된다. 우선 고객을 위한 소프트웨어로서 전자지갑이 있는데, 이 소프트웨어는 해쉬 체인을 생성하여 인증서를 발급받고 지불 처리를 수행하는 클라이언트(client)로서 구성된다. 전자지갑은 Windows NT 4.0에서 VC++ 5.0으로 구현되며 웹브라우저의 helper 프로그램을 이용하여 동작하게 된다. 중개인은 고객에게 인증서를 발급하는 인증서 발급 서버(server)와 상인의 정산요구를 수행하는 정산 서버로 구성되며, HP-UX 10.20에서 UNIX C로 구현된다.

상인은 고객의 지불을 처리하는 지불처리 서버와 정산할 때 사용되는 정산 클라이언트로 구성된다. 웹서버는 Solaris 2.5.1에서 Netscape 웹서버를 사용하고, 지불처리 서버는 gcc 2.8.0에서 CGI(Common Gateway Interface) 방법을 사용하여 구현된다. 중개인과 상인의 자료관리를 위해서 INFORMIX SERVER를 사용한다. 고객, 상인, 중개인 모두 인터넷에 연결되어 동작한다. 고객과

상인은 HTTP(HyperText Transfer Protocol) [18]을 이용한 웹 방식으로 연결되며, 고객/상인과 중개인은 TCP/IP(Transmission Control Protocol/Internet Protocol)[19, 20]로 연결된다. <그림 2>는 소액 지불 전자상거래 시스템의 전체 구성을 나타낸다.

4.2 고 객

고객의 전자지갑은 해쉬 체인을 만들며, 중개인과 TCP/IP로 연결하여 인증서를 발급받고 상인에게 지불을 수행하는 프로그램이다. 고객이 브라우저에 나타난 상품을 선택하면 상인의 웹서버는 CGI로 상인의 계정, 상품의 가격, 상품번호를 MIME(Multipurpose Internet Mail Extensions) type [18]과 함께 전송한다. MIME type과 상품정보를 받은 브라우저는 helper 프로그램 기법을 통해서 전자지갑 프로그램을 호출한다. 여기서 사용하는 해쉬 함수는 MD5로써 128비트를 사용한다[17].



〈그림 2〉 소액 지불 전자상거래 시스템 구성도

4.3 중개인

고객과 상인의 계정을 관리하며 고객에게 해쉬 체인에 대한 인증서를 발급해 주며 상인의 정산 처리를 한다. <표 1>은 상인의 계정과 정산에 사용될 체인의 정보를 저장하는 테이블 구조이다.

〈표 1〉 체인 정보

필드명	타 입	설 명	비고
CID	char(20)	인증서 고유번호	primary key
CUserID	char(20)	인증서 소유자	foreign key
CVendorID	char(20)	관련된 chain의 상인	
ChainLength	integer	체인의 길이	
ChainRoot	char(33)	w0	
Expire	char(11)	체인 유효 기간	
LastIndex	integer	마지막 정산된 인덱스	
LastToken	char(33)	마지막 정산된 체인	

4.3.1 인증서 발급 서버

고객에게 인증서를 발급해주는 서버이다. 고객의 계정 정보에서 인증서 발급을 요청한 체인의 금액이 신용한도액을 초과하지 않는지 조사한 후, 유효 사용기한을 첨부하여 인증서를 발급해준다. 그리고 신용한도액을 생성한 체인의 금액만큼 빼주어서 신용한도액을 유지한다. 인증서의 구조는 체인의 정보인 체인의 고유번호, 체인과 연관된 상인의 계정, 고객의 계정, 체인의 루트, 유효 사용기한을 가지는 CERTINFO와 그 내용의 변경, 위조를 막기 위해서 CERTINFO을 중개인이 전자서명한 CERTSIGN으로 구성된다. <그림 3>은 인증서의 구조체를 나타낸다. <그림 4>는 인증서를 생성하는 코드이다.

4.3.2 정산 서버

상인의 정산 요구를 처리해 주는 서버이다. 상인

```
typedef struct tagCERTINFO {
    char CID[ID_LEN];
    char VendorID[ID_LEN];
    char UserID[ID_LEN];
    TOKEN W_0;
    int ChainLength[TOKEN_LEN];
    char expire[11];
}CERTINFO;

typedef stuct tagCERTSIGN{
    int len;
    ungsigned char data[SIGN_MAX];
}CERTSIGN;

typedef struct tagCERTSTRUCT {
    CERTINFO info;
    CERTSIGN sign;
}CERTSTRUCT;
```

〈그림 3〉 인증서 구조

```
CERTSTRUCT cert;
unsigned char md[MD_LEN];
/* cert.info를 채운다.*/
...
/* MD5 function으로 cert.info에 대해 메시지
   다이제스트 만든다.*/
MD5((unsigned char*)&cert.info, sizeof(cert.in-
fo), md);

/* 메시지 다이제스트를 중개인의 비밀키로 암
호화 한다. */
PrivateEncrypt(md, &cert.sign);
```

〈그림 4〉 인증서 생성

이 정산을 요구한 토큰의 정당성을 검사하고 정산되어야할 금액인 현재 요청한 정산 토큰의 인덱스에서 마지막으로 정산된 토큰의 인덱스를 뺀 값에 해당하는 금액을 고객의 계정에서 상인의 계정으로 넣어준다. 그리고 마지막 정산된 토큰과 그 인덱스를 저장한다.

4.4. 상 인

4.4.1 지불처리 서버

CGI 형태로 동작하며 처음 거래하는 고객은 인

증서를 지불메시지와 같이 보내주는데 <그림 5>는 인증서가 조작되거나 변경되지 않았는지 검사하는 코드이다. 중개인의 전자서명을 체크함으로써 이 인증서가 중개인에 의해 작성되었다는 것을 확신할 수 있고 그 내용을 믿을 수 있다. 이 체인의 정보는 저장되어 다음 지불부터 계속 사용되게 한다.

지불 메시지는 체인 고유번호, 체인 및 인덱스인데, 체인 고유번호에 해당하는 체인의 정보를 읽어와서 그 체인이 유효한지, 이중 지출되었는지,

```
BOOL IsValidCert(CERTSTRUCT *cert)
{
   unsigned char sign_md[MD_LEN], org_md
   [MD_LEN];
   /*중개인의 공개키로 인증서의 사인부분을 복
     호화 한다. */
   PublicDecrypt(cert->sign, sign_md);
   /*인증성의 info 부분의 메시지 다이제스트를
     만든다.*/
   MD5((unsigned char*)&(cert->info), sizeof
   (cert->info), org_md);
   if(memcmp(sign_md, org_md, MD_LEN)==0)
      return TRUE;
   else
      return FALSE;
}
```

<그림 5> 인증서 확인

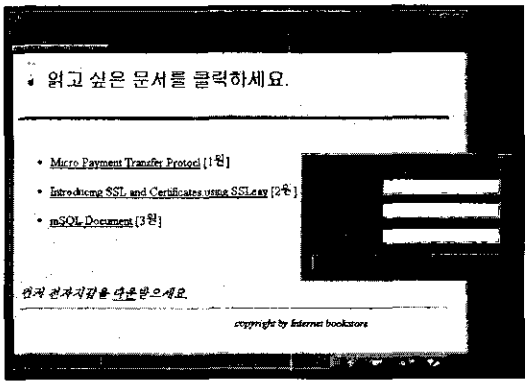
체인의 길이를 초과하였는지, 유효 사용기한이 지나지 않았는지를 체크한 후, 정당하다면 지불로 인정하고 서비스를 제공해준다.

4.4.2 정산 클라이언트

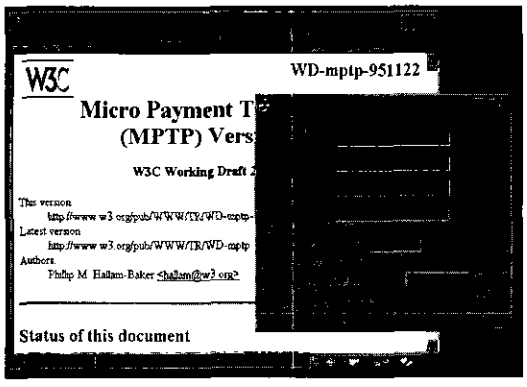
최근에 받은 모든 지불의 정보(체인 고유번호, 마지막으로 지불한 체인 및 인덱스 등)을 정산 서버로 보낸다.

4.5 실행 화면

<그림 6>은 읽고자 하는 문서를 클릭하여 전자지갑이 수행되고 새로운 체인을 생성하기 위해서 사용자의 ID와 패스워드 그리고 체인의 전체 금액을 입력받는 화면이다. <그림 7>은 해쉬 체인을 생성하여 문서를 웹브라우저에 받은 화면을 보여준다.



<그림 6> 해쉬 체인 생성 화면



<그림 7> 상품 구매 화면

5. 결 론

인터넷을 이용한 전자상거래는 날이 갈수록 발전하고 있으며, 그 수요도 급격하게 증대되고 있다. 본 논문에서는 인터넷을 이용한 소액 지불 프로토콜을 제안하고, 이를 기반으로 한 전자상거래 시스템을 설계 구현하였다. 소액 지불 전자상거래 시스템은 온라인 상품인 데이터베이스의 검색, 소프트웨어 판매, 전자신문 및 전자저널 서비스 등과 같은 상품 구매에 대해서 소액의 금액을 빈번히 지불해야 할 경우에 적합한 시스템이다.

또한 지불 금액이 소규모이므로, 시스템의 오류로 인한 피해가 다른 지불 시스템들보다도 적은 장점이 있다.

본 논문에서 제안하는 소액 지불 프로토콜은 PayWord 시스템을 기반으로 하여 설계되었다. 그러나, PayWord와는 달리, 고객은 인증서에 명시된 체인 길이 만큼의 상품 구매만 가능하므로, 상인은 영업일이 끝날 때마다 매번 중개인에게 정산을 요청할 필요가 없도록 설계되었다. 또한 체인의 유효성 및 이중 지출 등을 점검할 수 있다.

소액 지불 전자상거래 시스템에서의 고객 시스템은 Windows NT 4.0에서 VC++ 5.0으로 구현되었으며 웹브라우저의 helper 프로그램을 이용하여 동작된다. 중개인 시스템은 인증서 발급 서버와 정산 서버로 구성되며, HP-UX 10.20에서 UNIX C로 구현되었다. 상인 시스템은 지불처리 서버와 정산 클라이언트로 구성된다. 웹서버는 Solaris 2.5.1에서 Netscape 웹서버를 사용하였고, 지불처리 서버는 gcc 2.8.0에서 CGI 방법을 사용하여 구현되었다. 중개인과 상인의 자료관리를 위해서 INFORMIX 서버

를 사용하였다. 고객, 상인, 중개인 모두 인터넷에 연결되어 동작한다. 이를 위하여 고객과 상인은 HTTP를 이용한 웹 방식으로 연결되며, 고객/상인과 중개인은 TCP/IP로 연결되었다.

본 연구에는 다음과 같은 한계가 있다. 특히, 본 연구에서 제안하고 있는 프로토콜의 오버헤드(overhead)가 소액 상품을 구매하는데 문제가 없는가에 대한 이견이 제시될 수 있다. 본 논문에서 제안한 프로토콜의 오버헤드가 소액 상품 구매에 비해서 약간 크게 보일 수도 있으나, 제안된 프로토콜 동작의 대부분은 보안과 검증에 사용되고 있다. 소액이든 고액이든 거래규모에 상관없이 인터넷을 통한 상거래 과정에서는 필수적으로 보안과 검증 절차가 따라야 하므로 이러한 측면에서 보면 본 논문에서 제시하고 있는 프로토콜이 나름대로 의미를 갖는다고 믿어진다. 그럼에도 불구하고, 본 논문에서 제시하고 있는 프로토콜이 좀 더 효과적으로 적용되기 위해서는 오버헤드를 최소화 할 수 있는 연구가 지속되어야 할 필요가 있다.

참 고 문 헌

[1] 메타랜드, http://www.metaland.com.

[2] 박현동, "WWW Security", 5th WWW Workshop 자료집, pp 197-205.

[3] 성기윤, "인터넷 전자 지불 시스템의 현황", http://mis.cau.ac.kr/market/mis/ec/ec_re_paper/ec15.html.

[4] 현대백화점, http://www.hyundaidept.com.

[5] Dumas, A. "Programming Winsock," SAMS Publishing, 1996.

[6] Herzberg, A. and H. Yochai,MiniPay : Charging per Click on the Web.

[7] CyberCash, http://www.cybercash.com.

[8] Digicash, http://www.digicash.com.

[9] Young, E., SSLeay 0.9.0, ftp://ftp.psy.uq.oz.au/ pub/Crypto/SSL.

[10] First Virtual, http://www.fv.com.

[11] iKP Family of Secure Electronic Payment Protocols, http://www.zurich.ibm.com/Technology/security/extern/ecommerce.

[12] Lee, J.K., Analysis and Design of the Internet Based Payment System.

[13] Master Card and Visa, Secure Electronic Transaction Specification Version 1.0, May 1997.

[14] Millicent, http://www.millicent.digital.com.

[15] Hallam-Baker P.M., "Micro Payment Transfer Protocol(MPTP)," Nov. 1995.

[16] Rivest, R.L. and A. Shamir, PayWord and MicroMint : Two simple micropayment schemes, May. 1996.

[17] Rivest, R.L., The MD5 message-digest algorithm. Internet Request for Comments, RFC 1321, April 1992.

[18] Roberts, D., Internet Protocols : Handbook, Coriolis Group Books, 1996.

[19] Sidnie, F., TCP/IP : Architecture, Protocol, and Implementation with Pv6 and IP Security, McGraw-Hill Pub., 1997.

[20] Stevens, W.R., UNIX Network Programming : Networking APIs, Vol.1, 2nd. ed., Prentice Hall Pub., 1998.