PGP를 이용한 WWW 기반에서의 전자지불 프로토콜 개발

박 현 동[†] · 강 신 각[††] · 박 성 열[††] · 류 재 철[†††]

요 약

WWW은 미래의 쇼핑수단으로 자리를 잡아가고 있다. 하지만, 현재 사용되고 있는 지불형태는 구매자의 신용카드 정보가 평문으로 네트워크를 통해 전송되는 보안상의 문제점을 가지고 있다. 이러한 문제점들은 개인정보에 대한 침해 뿐만 아니라 경제범죄 등의 사회적 문제를 야기시킬 수 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 전자우편을 위한 보안도구로 개발된 PGP를 WWW에 접목시켜 WWW 암호통신을 구현하는 방법을 제시하고, 이 방법을 이용하여 새로운 전자지불 프로토콜인 SCCP를 설계, 제안한다. 제안하는 SCCP의 적합성을 확인하기 위하여 IBM에서 제시한 기준에 따라 검증한 결과, 본 논문에서 제시하는 방법은 안전한 전자지불 프로토콜로 판단된다.

A Development of WWW-based Electronic Payment Protocol using PGP

Hyun Dong Park[†] · Shin Gak Kang[††] · Sung Yul Park[††] · Jae Cheol Ryou[†††]

ABSTRACT

WWW has taken root in Internet as a means of future shopping. But, payment systems using todays show us several security vulnerabilities. The problem that plain shopper's credit card details are transferred in Internet is included. These risks can break out not only an infringement of private information but also economic crime. To solve these risks, we introduce the technique which implement the encrypted WWW communication using PGP. And we propose SCCP which is new electronic payment protocol. As a result of testing with criteria from IBM, we can find that SCCP is safe and secure electronic payment protocol.

1. 서 론

미국 국방성에서 개발한 TCP/IP 프로토콜을 기반으로 한 인터넷은 현재 수백만 대의 컴퓨터가 연결되어 있는 거대한 네트워크이다. 이러한 규모는 93년 이후부터 매년 2배씩 증가하는 추세이며 초기의 용도였던 교육, 연구용 등의 소수의 사용자를 위한 제한된 서비스의 범위를 벗어나 다양한 정보 전달의 수단으로 일반 사용자들의 사용이 크게 늘고 있다. 이러한 증가를 발생시킨 이유 중에는 인터넷에서 제공해 주는 여러가지 서비스들의 역할이 크지만 그 중에서도 WWW(World Wide Web)의 역할이 가장 주도적이라 할 수 있다. WWW은 1990년 Berners-Lee에 의해 처음으로 제안되었다. WWW은 초기에는 많은 관심을 받지 못하다가 Mosaic과 Netscape 등 사용자 인터페이스를 편리하게 설계하고 여러가지 형태의 자료를 손 쉽게 보여 주는 브라우저(Browser)의 출현으로 인하여 널리 퍼져 나가기 시작하였다. WWW의 편리

† 준 회 원:충남대학교 컴퓨터과학과 대학원 박사과정
†† 정 회 원:한국전자통신연구원 정보기술개발단
††† 종신회원:충남대학교 컴퓨터과학과
논문접수:1996년 12월 3일, 심사완료:1997년 3월 24일

한 사용자 인터페이스는 전문가들 뿐만 아니라 일반인들의 인터넷 사용을 증가시켰고, 결과적으로 인터넷 전체의 크기를 확대시키고 있다. 인터넷은 초기에는 전문가들이 주로 교육, 연구용으로 사용하였고, 그 결과로 인터넷에는 수 많은 정보들이 저장되게 되었다. 이러한 정보를 이용하고자 사람들이 인터넷에 모이게 되었고, 이제는 WWW을 도구로 삼아 인터넷을 상업적으로 사용하려는 단계로 접어 들고 있다[1].

WWW은 FTP(File Transfer Protocol), Telnet, Archie 등 기존의 인터넷 서비스와는 구별되는 편리한 사용자 인터페이스를 제공해 주고, 다양한 형태의 자료를 처리해 줄 수 있기 때문에 상업적 이용이 유리하다. 상점을 운영하는 입장에서 보면 WWW은 매우 매력적인 상점 운영 방식이다. 실세계에서 상점을 운영할 때에 필수적인 상점 건물 등이 필요하지 않고, 네트워크 기능을 갖춘 서버 컴퓨터만 있으면 상점을 개장할 수 있다. 또한 구매자의 입장에서도 쇼핑의 시간적 제약이나, 공간적 제약이 없다는 점에서 사용이 늘어날 것으로 보이고 있다. 이미 인터넷에는 많은 수의 상점이 개장되어 운영 중에 있다. 이러한 이유들로 인해 앞으로 인터넷의 가상 상점의 수는 그 수가 기하급수적으로 늘어날 것이라는 예측을 하고 있다.

하지만, 현재 운영 중에 있는 가상 상점들을 살펴 보면, 대금 지불 처리가 매우 위험한 방법으로 수행되고 있다. 대부분의 가상 상점들이 전자화폐보다는 사용에 있어 편리하고 구현이 용이한 신용카드를 이용하여 지불을 처리하고 있는 데, 신용카드의 번호, 비밀번호 등과 같이 비밀리에 처리되어야 할 정보들이 평문으로 네트워크를 통해 전송되고 있다. Yahoo 디렉토리에 등록되어 있는 약 300여 개의 가상상점들을 표본으로 조사한 통계의 결과를 보면 〈표 1〉과 같다.

〈표 1〉 인터넷 상점의 운영 형태
〈Table 1〉 Types of internet shop

운 영 형 태	비율
제품의 광고만을 인터넷을 사용하고 주문과 대금 지불은 상점에 직접 가서 처리하는 방법	16%
광고와 주문은 인터넷을 사용하고 대금지불은 구입자가 상점에 직접 가서 하는 방법	10%
광고, 주문, 지불이 모두 인터넷을 사용. 회원ID를 얻기 위해 신용카드 정보를 전송하고 대금지불 때에는 발급 받은 회원ID를 전송하는 형태	16%
광고, 주문, 지불이 모두 인터넷을 사용. 대금지불 때마다 신용카드 정보를 전송하는 형태	58%

〈표 1〉에서 보면 대금지불을 처리해 주기 위해 인터넷을 사용하는 상점은 전체 상점들 중에서 74%를 차지하고 있다. 그리고, 이 74%중에서 약 절반 이상이 안전하지 않은 방법으로 소비자의 신용카드 정보를 전송받고 있다. 보안을 고려하지 않았을 경우에 발생할 수 있는 문제점을 살펴 보면 첫째, 개인의 신용카드 번호와 같은 비밀 정보가 노출될 수 있다. 이는 개인정보의 노출에서 그치지 않고 이를 이용한 경제범죄 행위를 유발시킬 수 있다. 둘째, 구매자가 구매 사실을 부인하거나, 상점이 대금을 받지 못했다고 주장할 경우의 대비책이 전무하다. 셋째, 개인의 구매 정보가 노출됨으로 인해서 프라이버시 보장이 어렵다. 넷째, 불법적인 구매자 및 쇼핑 서버의 난립을 방지하기 어렵다.

본 논문에서는 이러한 WWW에서의 전자지불 처리에 예상되는 문제점들을 해결하기 위해 신용카드를 이용하는 새로운 전자지불 프로토콜인 SCCP(Secure Credit Card Payment)를 소개한다. SCCP는 NCSA(National Center for Supercomputing Applications)에서 설계한 WWW 암호통신 프로토콜인 PGP-CCI(Pretty Good Privacy-Common Client Interface)를 기반으로 구현되었다. 본 논문의 2장에서는 현재까지 이루어진 WWW 보안 관련 연구와 전자지불 프로토콜들을 살펴 보고, 3장에서는 SCCP의 프로토콜을 설명한다. 4장에서는 SCCP의 구현을 살펴 보고, 마지막으로 5장에서 결론을 맺기로 한다.

2. 기존의 연구

2.1 WWW 보안

인터넷에는 많은 연구 자료들이 저장되어 있는 데, 이들 중에는 대외비의 성격을 지닌 것들도 상당 수에 달한다. 어느 특정한 집단 안에서만 정보를 공유하고 그 외의 사용자들에게는 열람을 허락치 않는 정보들의 관리 방법이 필요하게 되었다. 이는 곧 WWW 서버로 전송되는 요청 메세지들 중에서 사용자 인증을 거친 요청에게만 문서를 보여줄 수 있는 확실한 사용자 인증 방법을 요구하게 된 것이다[2]. 또한 사용자

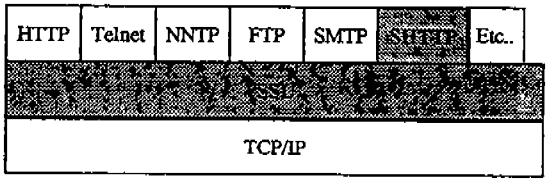
의 신원을 확인하고, 그 사용자에게 문서를 전송해 줄 때에도 문서의 기밀성(Confidentiality)을 보장해 주기 위해 HTTP(HyperText Transfer Protocol)의 암호화가 필요하게 되었다. 하지만, WWW의 근간인 인터넷이 사용하고 있는 TCP/IP는 본래 개방형 시스템을 기반으로 설계되었기 때문에 WWW은 기본적으로 보안성에서는 취약한 것으로 여겨진다[3]. 보안성이 취약한 WWW을 안전한 시스템으로 만들기 위한 현재까지의 연구상황을 살펴 보면 다음과 같다.

2.1.1 SHTTP(Secure HTTP)

1994년 EIT(Enterprise Integration Technologies)에서 발표한 SHTTP는 HTTP에 보안요소를 첨가한 형태의 프로토콜이다. SHTTP는 범용으로 사용될 수 있도록 설계되었으며 트랜잭션의 기밀성, 인증, 메세지의 무결성 등을 지원해 준다. 응용 레벨에서의 메세지 암호화를 통해 안전한 트랜잭션을 보장해 주고, RSA Data Security사의 공개키 암호 알고리즘을 이용하여 서버와 클라이언트가 공유하여야 하는 정보(세션키) 등을 암호화하여 전송한다[4].

2.1.2 SSL(Secure Socket Layer)

SSL은 "Netscape Communications"사에서 개발한 것으로 응용 프로토콜과 TCP/IP 사이에서 데이타의 암호화, 서버의 인증, 메세지의 무결성을 제공하고 클라이언트의 인증은 선택적으로 제공해 준다. SSL은 서버와 클라이언트 양쪽의 TCP/IP 연결을 위해서 "Handshake" 프로토콜을 수행한다. 이 결과로 양쪽은 암호화 통신에 합의하고, 암호화 통신과 인증에 필요한 값들을 준비한다. 이 단계가 지나면 SSL은 어



NNTP : News Network Transfer Protocol

SMTP : Simple Mail Transfer Protocol

(그림 1) SHTTP와 SSL의 위치
(Fig. 1) Location of SHTTP and SSL

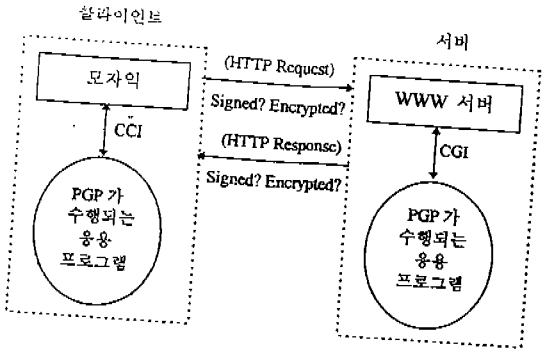
플리케이션 프로토콜에서 생성해 낸 바이트 열의 암호화와 복호화만 수행하게 된다. 이는 HTTP request와 HTTP response에 포함되는 모든 정보들(form의 데이타, access 인증자료,.)이 암호화되어 전송된다는 것을 의미한다[3].

SHTTP는 암호화가 응용 레벨에서 이루어지는 것에 비해, SSL은 (그림 1)과 같이 응용 레벨과 TCP/IP 레벨 사이에서 이루어진다.

SSL은 HTTP, NNTP와 같은 응용 프로토콜의 하위에서 동작하는것에 반해, SHTTP는 메세지-기반 접근 방법을 사용하여 응용 프로토콜에서 생성되는 메세지를 암호화한다. 하지만, 이 둘은 서로 배타적인 구조를 가지고 있는 것이 아니기 때문에 SSL의 상위에 SHTTP가 위치하는 구조로 각자가 가지고 있지 않은 부분들을 서로 보완하여 새로운 WWW 보안 메카니즘을 개발할 수 있다. SHTTP와 SSL은 모두가 공식적인 표준으로 지정되어 있는 상태가 아니므로 현재 EIT사와 Netscape Communication사는 공동연구를 통해 SHTTP와 SSL을 하나의 방법으로 통합하는 방법을 개발하여 공식 표준으로 만들기 위한 노력을 하고 있다.

2.1.3 PGP-CCI

이 외에도 NCSA에서 설계한 PGP-CCI가 있다. PGP는 Phil Zimmermann이 만든 응용 프로그램으로 전자우편의 보안도구로 사용하기 위해 만들어 졌다. PGP는 시스템에 독립적이고 공개키 인증이 현실적이어서 단지 전자우편이라는 제한된 분야에만 사용되지 않고, 암호화와 전자서명 등을 필요로 하는 분야에서는 얼마든지 응용될 수 있다. NCSA에서는 이런 PGP의 장점을 살려 WWW의 서버와 클라이언트에 PGP를 수행할 수 있는 응용 프로그램을 만들어 이들이 각각 서버와 클라이언트들과 통신을 할 수 있게 해줌으로써 HTTP의 암호화 전송을 이루고자 한다[2]. WWW 서버에서는 CGI(Common Gateway Interface) 기법을 이용하여 외부 프로그램과의 통신을 가능케 하고, 클라이언트에서는 CCI(Common Client Interface)를 이용하여 외부 프로그램과 브라우저 사이의 통신을 가능케 한다. 이 방법에서 사용하는 CCI 라이브러리는 현재 UNIX만을 지원해 주고 있기 때문에 브라우저는 UNIX용 Mosaic을 사용한다. NCSA에서



(그림 2) HTTP와 PGP가 정합된 형태
(Fig. 2) Model of HTTP including PGP

제안한 PGP-CCI를 그림으로 보면 (그림 2)와 같다.

안전한 WWW 통신을 위해 개발된 대부분의 프로토콜들이 HTTP와 HTML(HyperText Markup Language)의 수정을 요구하고 있는 것에 반해, 이 PGP-CCI 방법은 암호화와 전자서명 생성, 복호화와 전자서명 확인 등이 모두 외부 프로그램으로 수행되기 때문에, 기존의 HTTP와 HTML에는 아무런 수정을 요구하지 않는다.

2.2 전자지불 프로토콜

WWW에서의 전자지불을 위하여 선진국에서는 암호 메카니즘을 이용한 지불 시스템 개발을 서두르고 있으며, 그 결과로 전자화폐, 전자수표, 신용카드 및 지불 브로커 등의 방식이 소개되고 있다. 전자화폐나 전자수표를 사용하는 방법은 고도의 암호화 기술이 필요하며, 아직은 시기상조라는 견해가 지배적이다. 또 하나의 방법인 신용카드를 이용하는 방법은 이미 우리 사회에 신용카드의 사용이 대중화되어 있기 때문에 이를 이용하는 방법이 가장 현실적이라고 할 수 있다.

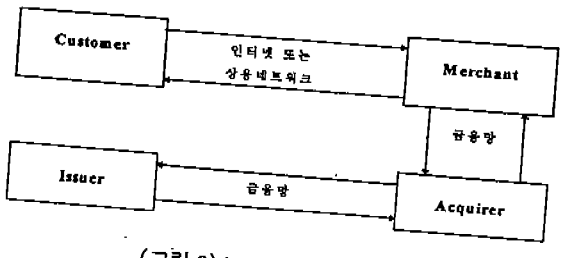
아직 공인된 표준이나 산업계 표준이 존재하지 않은 상황에서 여러 종류의 전자지불 시스템이 발표되어지고 있다. 이 중 몇 가지는 이미 구현이 되어 사용되고 있지만, 그 보다 많은 수가 아직 규격이나 계획만 발표하고 서비스는 시작하지 않고 있는 상태이다. 현재 실제로 상품을 구입할 수 있는 전자지불 시스템에는 CyberCash, First Virtual 등이 있다. iKP(Internet Keyed Payment)는 프로토콜만 발표해 놓은 채 아직 구현은 되지 않은 기술이다. 이들을 중심으로 전자지불 시스템의 종류와 상품구매 방법에 대하여 알아 보도록 한다.

2.2.1 iKP

iKP는 IBM에서 개발한 인터넷 전자지불 프로토콜로써 공개용으로 발표해 사람들의 의견을 듣고 있는 전자지불 프로토콜이다[5]. iKP라는 이름에서 i는 공개키를 소유하고 있는 당사자의 수를 나타낸다. 전체적인 프로토콜의 참여자는 4부분이다. 서비스를 사려는 사람(Customer), 서비스를 팔려는 사람(Merchant), 서비스를 팔려는 사람이 거래하는 은행(지불되는 금액이 입금되는 은행, Acquirer), 서비스를 사려는 사람이 거래하는 은행(지불되는 금액이 출금되는 은행, Issuer)이 그 당사자들이다. 이들 4부분의 참여자들 중에서 Acquire만이 공개키를 소유하고 있는 형태를 1KP라 하고, Acquirer와 Merchant가 공개키를 소유하고 있는 형태를 2KP, Acquirer, Merchant, Customer가 모두 공개키를 가지고 있는 형태를 3KP라 한다. 공개키를 소유하고 있다는 것은 전자서명을 발행할 수 있다는 것을 의미하며, 보안상의 측면에서는 3KP의 안전성이 가장 우수하다.

iKP의 특징은 구매자와 판매자는 인터넷이나 다른 상용 네트워크를 사용하고 있지만, 은행 사이의 거래는 이와 다른 금융망을 사용하고 있는 것이다. 이를 그림으로 표현하면 (그림 3)의 형태가 된다.

(그림 3)에서 보면, Customer와 Merchant 사이의 통신은 인터넷이나 상용 네트워크를 사용한다. 그러나, Merchant와 Acquirer사이의 통신, Acquirer와 Issuer 사이의 통신은 금융기관에서 사용하는 통신망을 사용하고 있다.



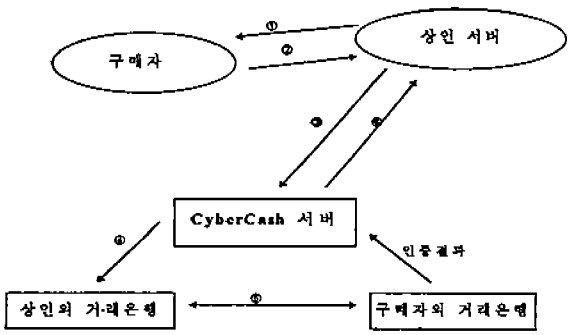
(그림 3) iKP의 네트워크 형태

2.2.2 CyberCash

현재 인터넷상에서 이루어지는 전자 거래에서 가장 일반적으로 사용되는 방법이 신용카드를 이용하는 것이다. 하지만 신용카드를 이용할 경우 신용카드 번호를 네트워크를 통해 상점으로 안전하게 보내는 방법, 불순한 의도를 가진 상점에 의해 신용카드 정보가 악용될 소지, 트랜잭션이 발생할 때마다 신용카드의 번호와 수신자 주소 등의 정보를 입력해야 하는 불편 등이 문제점으로 남는다.

CyberCash를 이용한 전자거래는 구매자, 상인, CyberCash Inc. 사이에서 이루어진다. 구매자는 CyberCash가 무료로 제공하는 CyberCash Wallet이라는 프로그램을 이용하여 CyberCash에 계정을 등록하고, 이 프로그램에 자신의 신용카드 정보를 기록한 후 이용한다. 상인은 CyberCash로부터 상인 자격을 인증받고, 인터넷의 WWW사이트에 자신의 상품을 선전하는 상점을 구축한다. 현재 CyberCash를 이용하는 크고 작은 상점은 CyberCash 홈 페이지에서 'Cool Places Shop'을 선택하면 이들의 목록을 볼 수 있다. CyberCash Inc.는 구매자와 상인을 서로에게 인증하고, 소비자의 신용카드 정보를 불순한 의도로부터 보호하며, 신용카드회사에게 상품 대금을 인출하여 상인에게 전달한다[6].

CyberCash가 동작하는 6단계는 (그림 4)와 같으며 각 단계별 설명은 아래와 같다.



(그림 4) CyberCash의 여섯 단계
(Fig. 4) Six steps in CyberCash

①구매자는 상인 서버에 접속하여 구입할 물건의 종류, 갯수, 배달될 주소 등을 지정한다. 상인 서버는 구매자가 입력한 것을 종합하여, 다시 구매자에게 보내준다.

②구매자는 상인이 보낸 값을 보고, 거래를 하기로 결정하면 화면에서 "Pay"버튼을 누른다. "Pay" 버튼을 누르면 Checkfree, Compuserver wallet 등이 구동되고, 거기서 지불에 사용할 신용카드를 선택한다. 그 후에 "OK"버튼을 누르면 지불정보가 상인에게 암호화되어 전송된다.

③상인은 수신한 암호문에 자신의 전자서명을 붙여 CyberCash서버로 전송한다. 이 때 상인은 구매자의 신용카드 정보 등을 알아낼 수 없어야 한다.

④CyberCash는 암호문을 복호화하여 지정된 라인을 통해 상인의 은행으로 지불내용을 전송한다.

⑤상인이 거래하는 은행은 구매자가 거래하는 은행으로 구매자에 대한 인증을 요청한다. 인증 요청에 대한 결과는 CyberCash서버로 전송된다.

⑥CyberCash서버는 인증결과를 상인에게 전송하고, 상인은 이를 구매자에게 전송한다.

2.2.3 First Virtual

First Virtual Holdings Incorporated사의 First Virtual(이하 FV)은 별도의 전용 프로그램이나 보안이 강화된 프로토콜을 이용하지 않고, 기본적인 WWW 브라우저와 전자우편만을 이용하여 전자지불 시스템을 구축했다는 점에서 다른 시스템과 구별된다[7]. 이는 일반 사용자에게는 지불 전용 프로그램이나 암호화 기법 등은 너무 복잡하고 불편하다는 이유에서 시작되었다. 따라서 이미 익숙한 WWW브라우저와 전자우편 만을 이용한 전자지불 시스템을 구현하였다. 그러나 특별한 암호화 기법을 사용하지 않기 때문에 신용카드 정보의 전달, 상품을 구입할 때 소비자가 계정을 개설한 사람임을 증명하는 방법 등의 보안상의 문제점을 내재하게 된다. FV에서는 이러한 문제를 전화, FAX, 전자우편을 이용하여 안전한 전자거래가 이루어질 수 있도록 한다.

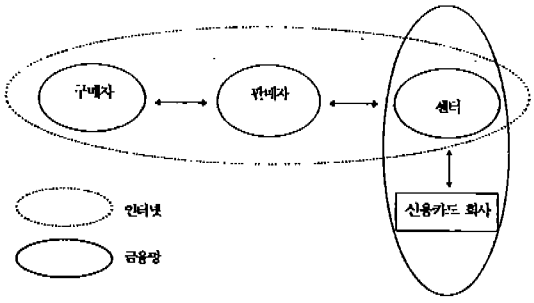
3. Secure Credit Card Payment 프로토콜

Secure Credit Card Payment(SCCP)는 전자우편 보안도구인 PGP를 사용하여 구현한 전자지불 프로토콜이다. PGP와 WWW의 접목을 위해서는 PGP-CCI

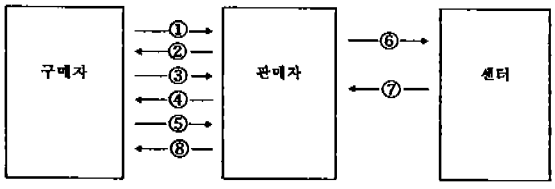
기법을 사용하였다. SCCP에서는 지불수단으로 신용카드를 사용하고 있으며, 지불보증 센터라는 기관을 두어 모든 거래에 대한 관리와 책임을 맡기고 있다. 신용카드를 지불 수단으로 사용한 이유는 현재 실세계에서의 지불수단으로 가장 널리 사용되고 있는 방법이고 전자화폐처럼 고도의 암호화 기술을 요구하지 않기 때문이다. 이 프로토콜은 WWW의 암호 통신이라는 문제를 해결한 기반 위에서 출발하므로, HTTP 메세지의 기밀성, 전자서명 등은 기본으로 제공되고 있다. 또한 사용자는 거래에 관한 모든 정보를 브라우저 화면 안에서 얻을 수 있도록 사용자의 인터페이스를 설계하였다.

3.1 SCCP의 지불 프로토콜

SCCP 프로토콜의 참여자는 3부분이다. 상품을 구입하려는 구매자와 상품을 팔려는 판매자, 그리고 거래를 관리하며, 사용자 인증을 수행하고 신용카드 회사에 명령을 내리는 센터가 각각의 당사자이다. 네트워크의 구성은 iKP에서 제시한 것처럼 센터와 신용카드 회사의 통신은 금융망을 사용하고, 구매자와 판매자 사이의 통신, 판매자와 센터 사이의 통신은 인터넷을 사용한다. 이는 (그림 5)와 같다.



(그림 5) SCCP의 전체 네트워크 구성도
(Fig. 5) Block diagram in SCCP



(그림 6) SCCP의 지불 프로토콜
(Fig. 6) Payment protocol in SCCP

SCCP 지불 프로토콜을 순서대로 나타내면 (그림 6)과 같으며, 각 단계에 대한 설명은 다음과 같다.

①지불페이지 요청

(그림 6)의 ①번 메세지는 지불페이지를 요청하는 단계로써 구매자는 쇼핑 도중에 구입하려는 물건을 선택한 후, 지불 페이지를 요청하게 된다.

②지불페이지 전송

지불 페이지를 요청받은 판매자는 지불페이지를 전송해 주는 데, ②번 메세지가 지불페이지의 전송이다. ①번 메세지와 ②번 메세지는 암호화 기법이 없는 단계로써 보통의 HTTP와 똑 같이 전송된다. 판매자는 ②번 메세지에서 판매자의 ID와 자신이 거래하고 있는 센터의 ID를 전송해 주어야 한다.

③구입 요청

EKS1[구매자ID‖코드‖갯수‖난수‖EKR 구매자[H(구매자ID‖코드‖갯수‖난수)]]‖EKU판매자[KS1]

③번 메세지에서 구매자는 구입하려는 물건의 코드, 갯수, 난수, 구매자ID를 판매자로 전송한다. 구매자ID는 판매자가 자신에게 문서를 암호화할 수 있도록 하기 위하여 자신의 공개키 ID를 알려 주는 것이고, 난수는 지불처리 후, 영수증을 받을 때에 자신이 신청한 거래에 대한 영수증 인지를 확인할 때 사용된다. 이 메세지에 구매자의 서명을 붙임으로써 구매자에 대한 사용자 인증이 이루어지도록 하고, 판매자만이 그 내용을 알아볼 수 있도록 암호화한다. 판매자의 공개키 ID는 ②번 메세지에서 hidden 태그를 사용하여 전송 받은 내용이다.

④가격 동의 요구

EKS2[가격동의서‖EKR판매자[H(가격동의서)]]‖EKU구매자[KS2]

가격동의서 : 판매자ID‖구매자ID‖코드‖갯수‖난수‖가격

③번 메세지를 수신한 판매자는 암호문을 복호화하고, 전자서명을 확인한다. 코드와 갯수를 이용하여 가격을 계산하여 가격동의서를 작성한다. 이 가격동의서에 판매자 자신의 전자서명을 추가한 후, 구매자만이 알아볼 수 있도록 암호화 하여 구매자에게 전송한다. 이 메세지에 판매자의 전자서명과 구매자의 전자서명을 추가함으로써 센터로 하여금 금액에 대해서 두 당사자가 합의를 했다는 증거로 삼는다

⑤가격 동의 확인

EKS3[가격동의서‖EKR판매자[H(가격동의서)]‖EKR구매자[H(EKR판매자[H(가격동의서)])]]‖EKU센터[KS3]

④번 메세지를 수신한 구매자는 복호화하고, 전자서명을 확인하여 가격동의서의 내용을 본다. 가격을 확인한 구매자는 판매자가 가격동의서에 붙여 놓은 전자서명에 자신의 전자서명을 추가하여 판매자가 붙인 전자서명과 함께 센터만이 알아볼 수 있도록 암호화하여 판매자에게 전송한다. 판매자의 전자서명에 구매자가 전자서명을 붙인 것은 가격에 대해 판매자와 구매자가 모두 동의했다는 증거로 활용된다.

⑥지불처리 요구

EKS4[가격동의서‖EKR판매자[H(가격동의서)]‖EKR구매자[H(EKR판매자[H(가격동의서)])]]‖EKU센터[KS4]

구매자로부터 ⑤번 메세지를 수신한 판매자는 ⑤번 메세지를 그대로 센터에게 전송해 준다.

⑦영수증 발급

EKS5[영수증‖EKR센터[H(영수증)]]‖EKU판매자[KS5]

영수증:지불성공여부‖센터ID‖판매자ID‖구매자ID‖코드‖갯수‖난수‖가격

센터는 ⑥번 메세지를 수신하여 복호화하고 구매자와 판매자의 전자서명을 확인하여 판매자와 구매자에 대한 사용자 인증을 수행한다. 전자서명 확인의 순서는 다음과 같다. EKR구매자[H(EKR판매자[H(가격동의서)])]을 복호화하여 구한 H(EKR판매자[H(가격동의서)]과 함께 전송된 EKR판매자[H(가격동의서)]의 해쉬값을 계산하여 비교한다. 두 값이 일치한다는 것은 구매자와 판매자가 동일한 가격동의서에 서명을 했다는 뜻이 된다. 두 값이 일치할 경우에는 EKR판매자[H(가격동의서)]을 복호화하여 구한 H(가격동의서)와 함께 전송된 가격동의서의 해쉬값을 구하여 비교한다. 두 값이 일치한다는 것은 가격동의서의 내용이 변조되지 않았다는 것을 의미하므로, 센터는 지불신청을 처리한다. 지불이 성공했는지의 여부를 나타내는 메세지와 센터ID, 구매자ID, 판매자ID, 코드, 갯수, 난수, 가격 등을 함께 하나의 메세지로 만들어 자신의 서명을 붙이고, 판매자만이 복호화할 수 있도록 암호화하여 판매자에게 전송한다.

⑧영수증 전송

EKS6[영수증‖EKR센터[H(영수증)]]‖EKU구매자[KS6]

센터로부터 영수증을 수신한 판매자는 센터의 전자서명을 구매자만이 복호화 할 수 있도록 암호화하여 구매자에게 전송해 준다. 사용된 센터ID, 판매자ID, 구매자ID, 코드, 갯수, 난수, 가격 등을 모두 비교해 봄으로써 지불처리를 신청한 거래에 대한 영수증임을 확인할 수 있다. ⑧번 메세지를 수신한 구매자도 센터ID, 판매자ID, 구매자ID, 코드, 갯수, 난수, 가격 등을 비교해 봄으로써 자신이 구입하려는 상품에 대한 지불 영수증임을 확인할 수 있다.

3.2 SCCP 프로토콜의 안전성

SCCP 프로토콜을 모든 통신에 대해 PGP를 이용하여 전자서명과 암호화를 구현함으로써 통신의 기밀성 및 송신 사실 부인 방지, 사용자 인증, 메세지 인증 등은 기본적으로 지원해 주고 있다. 그러므로, 이 단락에서는 지불 프로토콜이 가질 수 있는 위험성과 그 위험을 SCCP에서는 어떠한 방법으로 방지하고 있는 지를 알아 본다.

지불 프로토콜에 문제가 있다면 기밀성과 전자서명이 아무리 잘 구현되어 있다 할지라도, 제대로 지불 업무를 수행해 줄 수가 없다. 앞서 살펴 봤던 iKP에서 주장하는 지불 프로토콜이 가져야 할 보안기준들과 이를 SCCP에 적용한 결과를 알아 본다. 각 항목의 머리글로 들어가 있는 알파벳은 프로토콜 각각의 구성원의 입장에서 필요로 하는 요소를 뜻한다. 즉, A는 Acquirer로 판매자가 거래하는 은행의 입장에서 필요한 요소이고, M은 Merchant로 판매자의 입장에서 필요로 하는 요소를 나타낸다. C는 Customer로써 Buyer의 입장에서 필요로 하는 요소를 나타낸다. SCCP는 iKP와는 달리 센터가 신용카드 회사에 지불명령을 내리면 신용카드 회사는 현재의 금융망에서 사용하고 있는 방법대로 지불을 처리한다. 소개되는 기준들 중에서 앞의 8개는 지불 프로토콜에서 반드시 지켜져야 할 요소들이고, 뒤의 두 가지 요소 C5와 C6은 선택적으로 필요로 하는 요소들이다[5].

- A1:구매자에 의한 거래 인증(Proof of Transaction Authorization by Customer)

판매자의 은행 입장에서 구매자의 신용카드로부터

돈을 인출할 때는 이 트랜잭션이 실제로 신용카드의 주인이 허가한 것인 지를 반드시 알아야 한다.

SCCP에 이 기준을 적용해 보면, 신용카드 회사는 센터에서 내려 오는 명령을 근거로 하여 지불처리를 수행하게 되므로 결국, A1에 대한 확인작업은 센터에서 이루어진다. (그림 6)에서 구매자가 보내 주는 ⑤번 메세지가 확인작업을 수행해 준다. EKR구매자[H(가격동의서)]는 구매자ID에 구매자가 전자서명을 붙인 것으로 구매자 본인이 아니면 생성할 수 없는 메세지이므로 구매자가 신용카드의 주인이라는 것을 확인할 수 있다.

- A2:판매자에 의한 거래 인증(Proof of Transaction Authorization by Merchant)

판매자의 은행은 금액을 판매자의 계좌에 입금할 때, 이 계좌의 주인이 실제로 상품을 파는 판매자인 지를 반드시 알아야 한다.

이 기준을 SCCP에 적용해 보면, A2는 (그림 6)에서 판매자가 보내 주는 ⑥번 메세지를 이용해 확인할 수 있다. ⑥번 메세지의 EKR판매자[H(가격동의서)] 부분은 판매자ID에 판매자가 전자서명을 붙인 것으로 판매자 이외의 다른 사용자는 생성할 수 없다. 그러므로, 지불이 수행되어 금액이 입금되는 통장의 주인이 판매자라는 것을 증명할 수 있다.

- M1:Acquirer에 의한 거래 인증(Proof of Transaction Authorization by Acquirer)

판매자는 자신이 거래하고 있는 은행이 현재의 이 트랜잭션에 대해 적합성을 인증해 주기를 원한다. 이렇게 함으로써 나중에 판매자의 은행에서 트랜잭션의 적합성에 대한 의문을 갖지 않도록 한다.

SCCP에서는 판매자가 신청한 지불처리에 대해 모든 책임은 센터가 갖는다. 센터는 구매자와 판매자가 보낸 메세지들을 확인한 후, 신용카드 회사로 명령을 내리므로, 센터가 판매자가 보낸 메세지를 확인하여 신용카드 회사로 명령을 내렸다는 것으로 판매자는 자신의 지불 신청이 적합하다는 평가를 받았다는 것을 알 수 있다.

- M2:구매자에 의한 거래 인증(Proof of Transaction Authorization by Customer)

판매자는 자신이 거래하는 은행이 현재의 이 트랜잭션에 대해 적합성을 인정해 주었다 하더라도 Buyer에게서도 그러한 인정을 받아야 한다. 즉, 나중에 구매자가 그 거래에 대한 부정을 할 수 없게 한다.

SCCP에서 판매자는 자신이 계산한 금액에 대해 구매자의 확인을 받기 위해 (그림 6)의 ④번 메세지를 전송한다. ④번 메세지의 가격동의서에 구매자가 전자서명을 붙인다는 것은 구매자가 이 거래를 인정한다는 의미가 된다. 구매자의 전자서명이 붙은 ⑤번 메세지를 센터에 전송함으로써 판매자는 센터에게 자신의 지불요청이 구매자의 확인을 받은 것이라는 것을 증명할 수 있다.

- C1:지불요청은 구매자가 직접 해야 한다.(Unauthorized Payment is Impossible)

구매자가 스스로 거래를 신청해야 만이 구매자의 신용카드를 이용하여 출금할 수 있도록 해야 한다. 이는 구매자의 요청 메세지에 대한 replay 공격을 막아낼 수 있어야 한다는 것이다.

SCCP에서는 구매자가 상품을 신청하는 ③번 메세지와 가격에 동의한 ⑤번 메세지는 모두 구매자의 전자서명이 포함되어 있다. 구매자의 전자서명을 센터에서 확인하게 되므로 구매자 이외의 다른 사람은 정당한 거래를 신청할 수 없게 된다. 또한, 가격동의서에는 난수가 포함되어 있어 센터가 영수증을 발행할 때 포함시키는 난수와 비교하여 replay 공격을 방지한다.

- C2:Acquirer에 의한 거래 인증(Proof of Transaction Authorization by Acquirer)

구매자는 판매자의 은행으로부터 대금을 입금 받았다는 일종의 영수증을 받아 놓아야 한다. 이 영수증을 받아 놓음으로써 나중에 판매자가 대금을 지불받지 않았다고 할 경우와 또는 금액이 맞지 않는다고 주장하는 등의 분쟁에서 대금지불의 증거로 활용할 수 있다.

SCCP에서 구매자는 거래를 신청한 후, 영수증을 받게 된다. ⑧번 메세지에는 센터에서 발행한 전자서명이 붙어 있고, 이는 곧 이 거래가 센터의 확인하에 이루어졌다는 증거로 사용될 수 있다.

- C3:판매자의 인증서(Certification and Authentication of Merchant)

구매자는 지금 거래하려는 판매자가 은행으로부터 인증을 받은 판매자라는 증거를 원한다. 이러한 증거를 확인함으로써 구매자는 자신의 구매가 위조된 판매자에서 이루어지지 않고 있다는 증거를 확보하게

된다.

SCCP에서 판매자가 인증된 서버라는 사실에 대한 여부는 센터에서 전자서명의 확인작업으로 수행해 주고, 또한 구매자가 판매자의 공개키를 입수할 때에 공개키의 인증 여부로 확인할 수 있다. 센터는 정당한 판매자의 공개키에 자신의 서명을 붙여 공개한다. 만약, 구매자가 어느 판매자의 공개키를 얻었을 때 공개키에 센터의 서명이 붙어 있지 않으면 이는 정당한 서버가 아니라는 사실을 유추할 수 있다. 또한, 구매자가 만약 공개키의 인증 여부를 확인하지 않고 사용했다 할지라도, ⑥번 메세지를 받은 센터가 판매자가 정당하지 않다는 것을 발견할 수 있고, 이 때에는 지불을 처리하지 않게 된다.

• C4: 판매자가 발행한 영수증(Receipt from Merchant)

구매자는 판매자가 대금을 입금 받고 상품을 반드시 배달해 준다는 약속을 받아 놓아야 한다. 판매자가 금액을 입금 받고도 상품을 배달해 주지 않을 때에 사용할 수 있는 증거가 필요하다.

판매자는 센터에서 지불을 처리해 주어 입금을 받은 후에 상품을 구매자에게 배달하지 않을 수도 있다. 하지만, 지불보증 센터에는 판매자와 구매자의 전자서명이 붙어 있는 ⑥번 메세지가 저장되어 있으므로, 거래 자체에 대해 부정을 할 수는 없다. 만약 정해진 기간 내에 상품이 배달되지 않을 경우, 구매자는 센터의 협조를 얻어 대금 지불 내용을 확인할 수가 있다. 그러므로 판매자가 금액만 입금 받고 상품은 배달해 주지 않는 상황을 방지할 수 있다.

• C5: 기밀성(Privacy)

구매자는 자신의 구매 정보가 거래 당사자들 이외의 사용자들에게는 알려지지 않기를 원한다. 신용카드 정보와 같은 것은 당연히 암호화되어 전송되지만 그 이외의 정보 즉, 구매자가 어느 물건을 몇 개 신청했는 가와 같은 정보도 개인의 취향을 노출시킬 수 있기 때문이다.

SCCP에서 네트워크를 통해 전송되는 모든 내용들은 PGP를 이용해 암호화된 상태로 전송된다. 특정인이 어느 상품을 몇 개 신청했는 가와 같은 정보도 개인의 사생활이나 취향을 드러내기 때문에 이러한 정보들에 대해서도 기밀성을 보장해 줄 필요가 있다. SCCP에서는 암호화 모듈로 PGP를 사용하여 메세지들을 암호화해 주기 때문에 개인의 프라이버시를 지켜 준다.

• C6: 익명성(Anonymity)

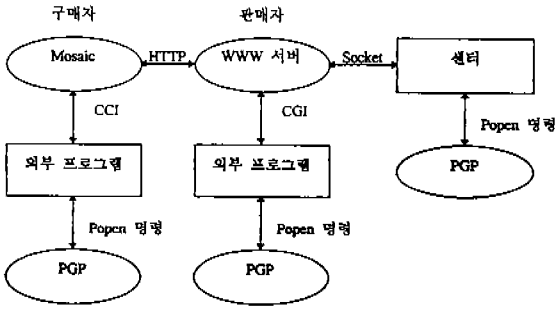
구매자는 자신이 보내는 정보에 대해 기밀성 이외에도 익명성을 보장받으려 할 때가 있다. 익명성은 상품을 판매하는 판매자의 입장에서 구매자가 누구인지를 모르게 하는 방법이 있고, 지불을 처리해 주는 쪽에서도 구매자가 누구인지를 모르게 해 주는 방법이 있다.

SCCP에서도 역시 구매자는 판매자에 대해 익명성을 제공받는다. 판매자가 알 수 있는 정보는 구매자가 신청한 상품의 종류와 갯수, 그리고 구매자의 ID이다. 하지만, ID만 사용해서 판매자가 구매자에 대한 정보를 알아낸다는 것은 불가능하다. 하지만, 센터에서는 구매자와 판매자의 ID, 신용카드 정보, 통장번호와 같은 정보를 저장하고 있기 때문에 구매자는 센터에 대해서는 익명성을 보장받지 못한다. 그러나, 센터에 대해서도 익명성을 지켜주려 할 경우에는 구매자에 대한 신분확인 작업과 지불처리가 불가능해지므로 센터에 대해서도 익명성을 보장받는 프로토콜은 불가능하다고 볼 수 있다. 그러므로, SCCP는 익명성을 보장받는다고 할 수 있다.

SCCP는 IBM에서 평가기준으로 제안한 10가지에 대해서 모두 만족하는 것으로 평가된다. 이는 iKP 프로토콜 중에서도 1KP와 2KP에서는 불가능하고 3KP에서만 실현이 가능한 기준들로써 이 사실은 SCCP가 iKP와 비교해 볼 때 보안적 측면에서 동등한 프로토콜임을 보여 준다.

4. SCCP의 구현

SCCP는 암호화 도구로 PGP를 사용하고 브라우저와 외부 프로그램 사이의 통신 메카니즘으로는 CCI를 사용한다. CCI는 NCSA에서 설계한 것으로 현재는 UNIX에서 수행되는 Mosaic만을 지원해 주고 있다. 그러한 관계로 SCCP의 구현은 UNIX용 Mosaic에서 수행되는 형태로 구현된다. 구매자는 UNIX에서 Mosaic을 사용하고, 판매자는 UNIX에 WWW 서버 프로그램을 구동시켜 쇼핑 서버를 만든다. 센터는 판매자의 CGI 프로그램과 socket을 생성시켜 통신을 한다. 이를 그림으로 나타내면 (그림 7)의 형태가 된다.



(그림 7) SCCP의 전체 구성도
(Fig. 7) Whole configuration in SCCP

구매자, 판매자, 센터에서 PGP를 사용하는 방법은 모두 popen 명령을 사용한다. PGP는 라이브러리화 되어 있는 것이 아니고, 하나의 독립된 프로그램 형태로 존재하기 때문에 외부 프로그램에서는 아래와 같은 코드를 실행하여 PGP를 수행시킨다.

```
Run_PGP(param)
char param[200];
{
   FILE * fp;

   fp = fopen("temp.sh", "w+");
   fprintf(fp, "export PGPPASS\n");
   fprintf(fp, "PGPPASS%s\n", "\"passphrase\"");
   fprintf(fp, "export PGPPATH\n");
   fprintf(fp, "PGPPATH = %s\n", "/home/hdpark/.pgp");
   fprintf(fp, "%s", param);
   fclose(fp);
   chmod("temp.sh", 0777);
   system("temp.sh");
   unlink("temp.sh");
}
```

PGP는 실행 도중에 사용자의 비밀키가 사용되어야 할 때는 사용자로 하여금 패스프레이즈(passphrase)를 입력하게 되어 있다. 그러나 센터나 판매자의 쇼핑 서버에서는 관리자가 계속 대기하여 입력작업을 해줄 수 없으므로 위와 같은 코드를 만들어 자동으로 패스프레이즈를 입력되도록 한다. 위의 코드가 수행되면 하나의 쉘이 기동되어 패스프레이즈가 자동으로 입력된다. 그 과정이 수행되면 패스프레이즈를 저장하고 있는 쉘은 자동으로 삭제된다.

4.1 CCI의 함수들

CCI는 Mosaic과 외부 프로그램 사이의 통신을 지원한다. CCI를 통해 외부 프로그램에서는 Mosaic이 방문하는 사이트의 URL을 알아낼 수 있고, 또는 특정한 URL을 지정하여 Mosaic이 그 곳을 방문하도록 명령할 수 있다. CCI의 대표적인 함수 몇 가지를 보면 다음과 같다[8].

4.1.1 Int MCCISendAnchor(serverPort, status, callBack, callBackData)

MCCISendAnchor 함수는 Mosaic이 새로운 URL을 방문할 때마다 그 URL을 외부 프로그램으로 전송해 준다. 즉, Mosaic에서 get 프로토콜이 수행될 때마다 그 프로토콜의 목적지를 알려 준다. serverPort는 외부 프로그램과의 통신로를 지정해 주는 곳이고, status는 이 함수의 사용상태를 나타낸다. Status를 MCCI_SEND_HANDLER로 해 놓으면 위의 설명과 같은 동작을 하고, 0으로 해 놓으면 MCCISendAnchor의 행동을 하지 않는다. callBack은 URL이 전송되었을 때 수행되어야 할 모듈을 지정해 준다.

4.1.2 Int MCCIGet(serverPort, uri, output, absRelative, additionalHeader)

MCCIGet함수는 Mosaic으로 하여금 uri에 지정해 주는 URL로 방문하게끔 한다. 이 함수는 외부 프로그램에서 어느 작업을 수행 중에 특정한 사이트로 Mosaic을 이동시키는 데 효율적이다. Output 파라메터는 새로운 URL을 방문할 때, 새로운 Mosaic을 기동시킬 것인지 아니면 현재의 Mosaic으로 방문할 것인 지를 결정해 준다. absRelative는 지정해 주는 URL이 절대경로인지 상대경로인 지를 알려 준다. additionalHeader는 추가적인 HTTP 헤더를 추가할 수 있는 파라메터인데 현재까지는 정의되어 있는 것이 없다.

4.1.3 Int MCCIPost(serverPort, uri, contentType, data, dataLength, output)

MCCIPost는 Mosaic으로 하여금 url에 지정된 서버로 data를 포스트(post)하게 한다. 이 함수는 암호

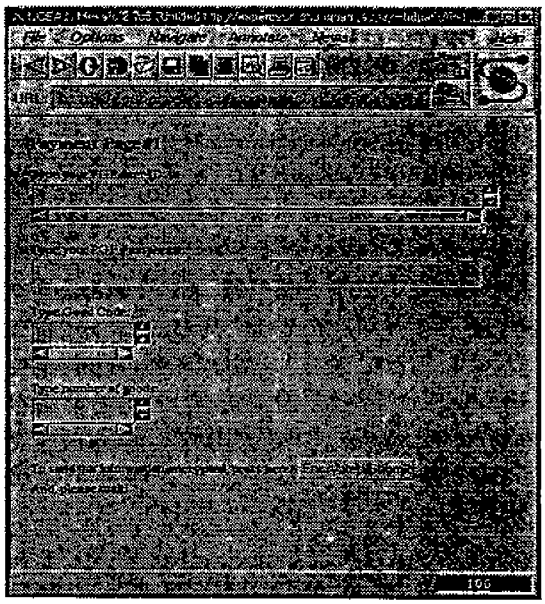
화된 query string을 data에 스트링으로 받아 서버의 CGI 프로그램으로 전송할 때 사용한다. output은 CGI 프로그램의 결과를 새로운 Mosaic을 실행시켜 보여줄 것인지, 아니면 현재의 Mosaic 화면에서 보여줄 것인지를 지정하는 파라메터이다.

4.1.4 Int MCCISendOutput(serverPort, mimeType, callBack, callBackData)

MCCISendOutput은 Mosaic으로 어느 특정한 Content-type의 메세지가 왔을 때 그것을 처리해 줄 수 있는 모듈로 메세지를 보내 준다. mimeType은 처리해 주려고 하는 Content-type을 지정해 주는 파라메터이다. 이 함수는 서버가 보낸 암호문을 받아 복호화하는 데에 사용된다.

4.2 Mosaic과 PGP의 연결

(그림 7)에서 외부 프로그램과 Mosaic의 연결은 CCI가 담당한다는 것을 알 수 있다. 구매자는 상품의 신청을 위해 Form 페이지를 전송 받는 데, 그 형태는 (그림 8)과 같다.



(그림 8) 상품을 신청하는 페이지
(Fig. 8) Goods request page

이 지불 페이지의 소스코드에는 다음과 같은 부분이 포함되어야 한다.

〈INPUT TYPE="hidden" NAME="PGP-POST" VALUE="yes"〉

〈INPUT TYPE="hidden" NAME="PGPSERVERID" VALUE="Shopserver"〉

"hidden" 태그는 브라우저 화면에는 보이지 않는다. SCCP에서 "hidden" 태그를 사용하는 이유는 query string을 판매자만이 볼 수 있도록 암호화하려 할 때에 필요한 정보 즉, 판매자의 공개키 ID 등을 "hidden" 태그를 이용하여 전달받을 수 있기 때문이다. "hidden" 태그에 의한 변수들은 "submit"버튼을 클릭하여 query string을 전송할 때에는 다른 변수들과 함께 전송된다. 특히, 서버에게 전송하는 방법으로 "get"을 사용하면 URL 뒤에 붙어서 전송된다. CCI의 함수들 중에서 MCCISendAnchor는 Mosaic이 "get"을 수행할 때마다 URL을 외부 프로그램쪽으로 보낼 수 있다. 이때 MCCISendAnchor 함수의 파라메터 중에서 status를 MCCI_SEND_ANCHOR로 선언해 주어야 query string이 서버쪽으로 전송되지 않는다.

외부 프로그램에서 query string을 암호화하여 서버쪽으로 보내 줄 때는 MCCIPost 함수를 사용한다. 이 함수는 보통의 Form페이지에서 query string을 post해 주는 것과 같은 역할을 수행한다.

서버쪽에서 전송해 주는 암호문은 MCCISendOutput 함수를 이용하여 처리한다. MCCISendOutput함수는 특정한 Content-type의 메세지가 입력될 때 메세지를 특정한 모듈로 보내주는 기능을 갖는다. 구매자와 판매자가 Content-type을 서로 지정해 놓으면 이 함수를 사용하여 수신한 암호문을 복호화하고 전자서명을 확인할 수 있다.

복호화된 내용을 Mosaic으로 보려 할 때에는 복호화된 내용을 임시화일에 저장시켜 놓고, local화일을 보는 방법을 사용해야 한다. 만약 보통의 URL을 다루듯이 "http://esperosun.chungnam.ac.kr/"과 같은 방법을 사용하면 복호화된 화일의 내용이 네트워크를 흐르게 되므로 암호화 작업의 의미가 없어지게 된다.

4.3 CGI프로그램과 PGP의 연결

CGI프로그램은 메세지의 입력/출력 형식이 이미 정의되어 있기 때문에 구매자의 외부 프로그램에서

필요로 했던 CCI기법과 같은 것은 따로 필요로 하지 않는다. 단지, PGP는 화일 단위로 수행되기 때문에 입력된 스트링을 화일에 받는 작업과 메세지의 출력을 위해 화일의 내용을 스트링으로 변환시켜 주는 작업이 필요하다.

CGI프로그램과 센터와의 연결은 socket을 이용해 구현한다. 일반적인 socket사용법과 다른 부분은 없고, 여기서도 역시 read한 스트링을 화일에 받아서 처리해야 하고, 암호화된 결과를 write해 줄 때는 스트링으로 변환시켜 주어야 한다.

4.4 센터에서의 지불 요구 내용 확인

구매자와 판매자가 기입한 내용들을 복호화하여 내용을 확인하는 센터는 내용들을 인식하기 위해 파싱작업을 해야 한다. 다른 메세지들은 단순한 파싱작업으로 가능하나, (그림 6)의 ③번 메세지에 대한 파싱작업은 약간 다르다. 이유는 판매자가 보내 준 ②번 메세지는 Form페이지의 형태를 취하고 있기 때문이다. ③번 메세지는 ②번 메세지를 그대로 놓고 구매자의 전자서명을 첨가한 것이다. 결국, ③번 메세지의 전자서명을 계속 확인하다 보면, Form 페이지의 소스코드가 나온다. 그래서, 센터에서는 내용을 확인하기 위해 단순한 "Name=Value"형식의 파싱이 아닌 Form 페이지의 소스를 파싱해야 한다.

5. 결 론

WWW이 쇼핑수단으로 본격화되고 있지만 지불수단으로 사용되고 있는 방법들은 보안상의 많은 문제점을 내포하고 있다. 안전하지 않은 전자지불 프로토콜의 사용으로 예상할 수 있는 문제점으로는 개인 정보의 유출과 경제범죄 발생을 예상할 수 있다. 이는 더 이상 간과할 수 없는 것으로 인식되어 여러 회사에서 전자지불 프로토콜의 개발을 위해 많은 투자를 하고 있다.

이러한 상황에서 WWW 기반의 전자지불 프로토콜로 개발된 SCCP는 신용카드를 이용한 전자지불 방식을 채택하였다. 이는 신용카드를 이용한 방식이 이미 사람들의 생활에 관례화되어 있어 구현과 사용이 편리하기 때문이다. 또한, SCCP는 구매자가 상품의 신청에서부터 거래의 결과까지를 하나의 브라우저 안에서 수행할 수 있는 형태의 편리한 인터페이스를 설계하였다. SCCP는 암호화 모듈로 사용한 PGP의 특성대로 인터넷에서 전송되는 모든 메세지들에 대해서 전자서명과 암호화 작업을 수행해 준다. 전자서명의 사용으로 대금지불 메세지에 대한 송신부인 및 메세지 인증, 사용자 인증이 가능해져 전자지불 프로토콜에서 요구하는 모든 사항들을 만족시켜 주고 있다.

참 고 문 헌

[1] 권도균, "WWW 보안과 전자화폐," http://wsp.nextel.net/seminar/etc/wwwpay.html

[2] Adam Cain, "CCI-based Web Security," 4th International WWW Conference, 1995.

[3] Hickman, K. and T. Elgamal, "The SSL Protocol," ftp://ietf.cnri.reston,va.us/internet-drafts/draft-hickman-netscape-ssl-01.txt

[4] Rescorla, E and A. Schiffman, "The Secure HyperText Transfer Protocol," http://info.internet.isi.edu/in-drafts/files/draft-ietf-wts-shttp-00.txt

[5] Mihir Bellare, "iKP:A family of secure electronic payment protocols," http://www.zurich.ibm.ch/Technology/Security/extern/ecommerce/iKP_overview.html

[6] CyberCash사, "The Six Steps in a Secure Internet Credit Card Payment," http://www.cybercash.com/cgi-bin/vdkw_cgi/x8eb907a8-4285/Search/4982900/1

[7] FV사, "First Virtual Overview," http://www.fv.com/info/overview.html

[8] NCSA, "Application Programmer's Interface:for the NCSA Mosaic Common Client Interface," http://www.ncsa.uiuc.edu/SDG/Software/Xmosaic/CCI/cci-api.html

박 현 동

1995년 충남대학교 전산학과(학사)
1995년 충남대학교 대학원 컴퓨터과학과(석사)
1997년~현재 충남대학교 대학원 컴퓨터과학과 박사과정
관심분야:네트워크 보안 시스템, 암호학

강 신 각

1984년 충남대학교 전자공학과(학사)
1987년 충남대학교 대학원 전자공학과(석사)
1984년~현재 한국전자통신연구원 책임연구원
관심분야:멀티미디어 통신, 고속통신, 정보보호

박 성 열

1977년 연세대학교 전자계산학과(석사)
1982년 Univ. of Florida 산업공학과(석사)
1987년 Auburn Univ. 산업공학과(박사)
1973년~1978년 한국과학기술원 연구원
1987년~현재 한국전자통신연구원 정보기술개발단장
관심분야:분산처리, 정보보호, 인터넷 응용

류 재 철

1985년 한양대학교 산업공학과 졸업(학사)
1988년 Iowa State University 전산학과(석사)
1990년 Northwestern University 전산학과(박사)
1991년~현재 충남대학교 컴퓨터과학과 조교수
관심분야:통신망 관리, 컴퓨터 및 통신 보안, 분산처리