通信情報保護學會論文誌
第 10 券 第 2 號, 2000. 6

이동컴퓨팅 환경에서 트랜잭션의 원자성을 보장하는 소액지불시스템

김 상 진*, 성 주 환**, 오 희 국*

Atomic Micropayment System for Mobile Computing Environment

Sangjin Kim*, Juhwan Sung**, Heekuck Oh*

요 약

이동컴퓨팅 환경은 기존의 컴퓨팅 환경과는 달리 호스트의 이동성과 무선 통신의 특성을 고려하여야 한다. 기존의 지불시스템들은 이와 같은 특성을 고려하지 않고 설계되었기 때문에 이동컴퓨팅 환경에서 이를 효율적으로 사용할 수 없다. 본 논문에서는 이동컴퓨팅 환경에서 소액지불을 위한 새로운 전자지불시스템 AMPS(Atomic MicroPayment System)를 제안한다. 이 시스템은 고객인 이동 호스트의 연산 부담과 무선 통신 메시지의 수를 최소화하도록 설계되었다. AMPS는 고객의 요청이 들어오면 대행 서버를 통해 고객을 대신하여 유선망에서 판매자를 상대로 상품 거래를 대행하여 준다. AMPS는 소액의 디지털 상품 구매를 위한 지불시스템으로 지불 트랜잭션의 원자성(atomicity)을 보장한다. 또한 판매자와 제 3자로부터 고객의 익명성과 대행 서버가 구매 상품 내역에 대해 알 수 없도록 구매 상품의 비밀을 보장한다. AMPS 프로토콜의 안전성과 효율성에 대해서도 분석하였다.

ABSTRACT

In order to provide an efficient payment service in mobile computing environment we must consider the characteristics of wireless communication and the mobility of hosts. Since current payment systems did not consider or aimed for such environment they inherently possess some properties that are inefficient for mobile computing. In this paper we propose a new micropayment system AMPS(Atomic MicroPayment System) for mobile computing environment. AMPS reduces the computation load of mobile hosts by moving operations that are normally perform by the client to the static portion of the network. In AMPS, clients request goods to a trusted AMPS server using only one wireless message and it can also be disconnected while AMPS server deals with the merchant. Thus reducing communication cost and power consumption. AMPS also protects user privacy by hiding the details of the payment to the AMPS server. AMPS server can also provide client anonymity to merchants. Another advantage of using AMPS server is atomicity. AMPS server provides transaction atomicity by checking goods and money before forwarding to client or merchant and preserving necessary information for future dispute. We have also analyzed security and efficiency aspect of AMPS protocol.

keyword : *electronic payment system, atomicity, mobile computing*

Ⅰ. 서 론

최근 인터넷을 이용한 상거래는 인터넷 시장의 확산과 보안 기술의 발전으로 급속히 활성화되고 있으며 수많은 인터넷 쇼핑몰이 등장하고 있다. 인터넷은 전통적인 유형의 상품을 판매하는 장으로서의 역

* 한양대학교 전자계산학과({sangjin, hkoh}@cse.hanyang.ac.kr)
** (주)동성정보통신(jhsung@ipo.co.kr)

할뿐만 아니라 전자신문, 전자잡지, 사진, 그림, 만화, 주식정보, 게임, 음악, 비디오와 같은 소액의 디지털 상품을 판매하기 위한 수단으로 활용가치가 매우 높다. 하지만 현재 인터넷 상거래에서 가장 널리 사용되고 있는 신용카드는 처리비용과 수수료 문제 때문에 액수가 작은 정보 상품의 지불 수단으로는 적합하지 않다. 따라서 소액의 디지털 상품을 구매하기 위한 소액지불시스템과 인터넷에서 유통시킬 수 있는 안전한 전자화폐가 요구되며, 비용을 지불하지 않고는 전달될 상품을 열람할 수 없도록 하는 상품의 보호 장치가 요구된다.

전자화폐에 대한 연구개발은 1982년 David Chaum의 온라인형 전자화폐 시스템이 처음 등장한 이래 전자화폐의 여러 가지 요구조건을 만족시키는 많은 방식과 시스템이 제안되고 있다.[1] 각 지불시스템을 분석하는 특징에는 일관성(consistency), 익명성(anonymity), 확장성(scalability), 원자성(ato-micity), 용인성(acceptability), 경제성(econo-mic) 등이 있다. 이 중 최근에 이슈로 등장하고 있는 것은 원자성이다.

원자성이란 데이터베이스 트랜잭션에서 말하는 원자성과 동일한 개념으로서 프로토콜이 정상적으로 완료되어 참여자가 모두 원하는 효과를 얻거나 아니면 프로토콜 시작 전과 동일한 상태를 유지하여야 한다는 것이다. 기존 시스템들은 전자화폐가 실물화폐와 동일한 기능들을 가지기 위한 노력에만 집중한 나머지 가장 기본적으로 보장하여야 하는 원자성에 대해서는 거의 고려하지 않고 있다. 그러나 원자성이 보장되지 않는 지불시스템을 사용하면 고객은 금액을 지불하였음에도 불구하고 상품을 받지 못할 수도 있고 판매자는 상품을 전달하였음에도 불구하고 대금을 받지 못할 수도 있다. 이와 같은 상황은 참여자 시스템에 문제가 발생하여 일어날 수도 있고 통신상의 문제로 일어날 수도 있다. 특히 이동컴퓨팅 환경에서는 이동 호스트의 제한된 자원 문제와 무선 통신의 특성인 간섭과 잦은 단절(disconnection)로 인해 트랜잭션의 원자성을 보장하는 것이 매우 중요하다.

이동컴퓨팅 환경에서 지불 트랜잭션의 처리는 유선 환경과는 또 다른 많은 제약 조건을 가지고 있다.[2] 이동 호스트는 프로세스 속도, 메모리 크기, 디스크 용량과 같은 자원 측면에서 유선 환경의 시스템들보다 상대적으로 빈약하며, 배터리로 운영되므로 전력이 제한되어 있다. 따라서 소프트웨어 측면에서는 이동 호스트 내에서 수행되는 연산을 프로세스 속도와 전력 소모를 고려하여 저렴한 연산으로 구성해야 하며, 이동 호스트의 연산 부담을 가능하면 유선 환경에 있는 시스템으로 옮기는 것이 바람직하다. 또한 이동 호스트의 통신 수단인 무선 통신은 유선 통신에 비해 환경에 간섭을 많이 받으며 단절될 확률이 높다. 이를 극복하기 위해 무선 통신 메시지의 수와 크기를 줄이는 것이 효과적이며 오랜 시간 동안 온라인을 유지하지 않고도 서비스를 받을 수 있도록 하여야 한다.

이 논문에서는 이와 같은 이동컴퓨팅 환경을 고려하여 이 환경에서 효율적이고 안전하게 소액의 정보 상품을 거래할 수 있는 지불시스템 AMPS(Atomic MicroPayment System)을 제안한다. 이 시스템을 사용하는 고객은 하나의 무선 메시지만을 전송하면 거래가 완료되도록 설계하여 무선 통신 메시지의 수를 최적화하였으며, 오랜 기간 동안 온라인으로 유지하지 않아도 정보 상품을 구매할 수 있도록 하였다. 이를 위해 AMPS는 유선 환경에서 고객을 대신하여 거래를 대행하여 주는 대행서버를 사용한다. 이와 같은 서버의 사용을 통해 이동 호스트의 연산 부담과 통신비용을 줄여주는 한편 지불 트랜잭션의 원자성을 보장하고 있다.

지불 트랜잭션에 원자성은 Doug Tygar의 분류에 따라 금전의 원자성(money atomicity), 상품의 원자성(goods atomicity), 보증 배달(certified delivery)로 나눌 수 있다.[3] 금전의 원자성이란 전자화폐가 교환되는 트랜잭션에서 기본적으로 보장하여야 하는 것으로서 전자화폐가 통신 채널로 이동할 때 화폐가 소멸되거나 금액의 변동이 발생할 수 없도록 보장하는 것을 말한다. 상품의 원자성은 금전의 원자성이 보장되고 추가로 가격을 지불하고도 상품을 받지 못하거나 상품을 전달하였는데 상품 대금을 받지 못하는 경우가 없도록 보장하는 것을 말한다. 보증 배달은 상품의 원자성이 보장됨은 물론 수신한 상품이 원래의 상품과 동일하도록 보장하는 것을 말한다. 위 세 가지 원자성은 지불과정에서 필요한 원자성이다. 그러나 전체 지불시스템에서는 여기에 전자화폐 인출과 입금 과정의 원자성도 보장하여야 한다.[4] AMPS는 최상위 개념의 원자성인 보증 배달(certified delivery)을 보장한다. 또한 AMPS에서는 기존 전자화폐를 사용한다고 가정하고 있다. 즉, 기존 전자화폐의 인출 프로토콜을 사용하여 전자화폐를 인출한다고 가정한다. 그러므로 본 논문에서는 지불시스템의 인출 과정은 고려하지 않고 있으나 자체

개발한 입금 프로토콜을 사용하며, 입금의 원자성을 보장한다.

AMPS는 고객이 사용할 이동 호스트의 연산 부담을 줄이기 위해 전력 소모가 많이 요구되는 공개키 암호 연산의 횟수와 공개키로 암호화되는 데이터의 크기를 줄였으며, 가능하면 대칭형 알고리즘이나 해쉬 알고리즘을 사용하여 소액지불 처리비용이 상품 가격보다 낮도록 경제성도 함께 고려했다. 이 외에도 소액 정보 상품의 특성과 원자성을 보장하기 위해 정보 상품을 암호화하여 교환하며, 판매자와 제 3자로부터 고객의 익명성과 대행 서버가 구매 상품 내역에 대해 알 수 없도록 구매 상품의 비밀을 보장한다.

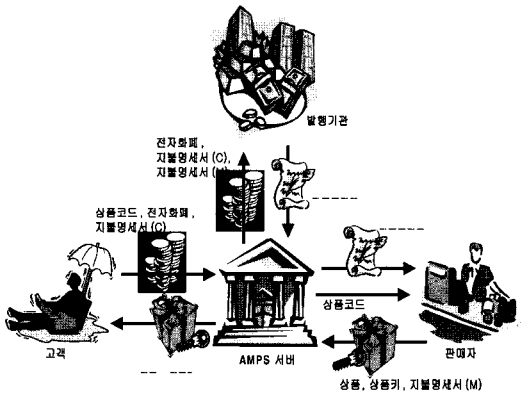
본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 AMPS에서 사용하는 가정과 각 참여자의 역할을 설명하고 AMPS에서 사용하는 프로토콜들을 자세히 기술한다. 3장에서는 제안하는 시스템의 안전성을 분석하고 기존 시스템과 비교한다. 끝으로 4장에서는 결론을 맺고, 향후 연구방향을 제시한다.

II. 지불시스템 AMPS

본 장에서는 AMPS에 참여하는 참여자의 종류와 역할, AMPS에서 사용하는 가정, 그리고 AMPS에서 사용하는 각종 암호프로토콜(지불, 정산, 상태질의 등)을 상세히 설명한다.

2.1 참여자

본 지불시스템에 참여하는 참여자로는 그림 1에 나타난 것처럼 인증기관(certification authority), 발행기관, AMPS 서버, 고객, 그리고 판매자가 있다. 각 참여자의 역할은 다음과 같이 정의한다. 인증기관은 판매자와 AMPS 서버에게 인증서(certificate)를 발급해주는 공인 기관이며, 발행기관은 전자화폐를 발행하고 고객과 판매자의 계좌를 관리하는 금융기관이다. AMPS 서버와 판매자는 각각 인증기관으로부터 인증서를 발급 받아야 서비스를 제공할 수 있다. AMPS 서버는 고객과 판매자 사이의 중재자 역할을 하며 유선망에 위치한 서버로서 고객으로부터 구매 요청을 받아서 고객을 대신하여 판매자 사이트에 접속하고 거래를 대행한다. 고객은 이동 호스트를 이용하여 디지털 상품을 구입하고자 하는 일반 사용자이며,



〔그림 1〕 지불시스템 AMPS

이를 위해 고객은 먼저 판매자 사이트에 방문하여 구입할 상품을 결정하고, AMPS 서버에게 구입할 상품과 상품 구매 대금을 제공한다. 판매자는 고객이 구입하려는 상품을 AMPS 서버에게 제공하고 상품 구매 대금을 받는다.

2.2 가정

본 시스템은 다음과 같은 몇 가지 가정을 기반으로 설계하였다. 첫째, 신뢰할 수 있는 공개키 기반 구조(public-key infrastructure)가 존재한다고 가정한다. 즉, 고객이 신뢰할 수 있는 공인 인증기관이 있으며, AMPS 서버와 판매자는 공인 인증기관으로부터 인증서를 발급 받아야 서비스를 제공할 수 있다. 또한 판매자와 서버의 인증서를 얻을 수 있는 디렉토리 서버가 존재한다고 가정한다. 따라서 고객이 거래를 하기 위해서는 해당 판매자와 AMPS 서버의 인증서를 가지고 있어야 하며, 각 인증서의 유효성을 확인하기 위해 필요한 공인 인증기관의 인증서를 가지고 있어야 한다.

둘째, 고객은 지불에 필요한 전자화폐를 발행기관으로부터 인출하여 이동 호스트에 가지고 있는 상태라고 가정한다. 본 시스템에서는 Ecash, PayWord, Micro-Mint와 같은 이중사용 여부를 발행기관이 확인하여야 하는 금전적 가치를 지닌 안전한 온라인 전자화폐를 사용한다고 가정한다.[5,6] 목표로 하는 소액지불 환경에 온라인 전자화폐의 사용이 적합하지 않다고 생각할 수 있다. 온라인 화폐의 문제는 판매자가 거래마다 은행에게 접속하여 확인하여야 하는 불편함과 이것에 따른 추가적인 비용이 소요되기 때문이다. 그러나 본 시스템에서는 판매자가 화폐를 은행에게

제시하여 확인하는 것이 아니고 AMPS 서버가 대신 확인하여 주므로 기존 환경과는 차이가 있다. 즉 판매자와 고객에게는 온라인의 사용에 따른 불편함이나 추가 비용이 소요되지 않는다. 또한 지금까지 제시된 대부분의 전자화폐는 화폐를 온라인으로 또는 오프라인으로 지불하는 것과 상관없이 이중사용에 대한 확인은 발행기관에서 유지하고 있는 데이터베이스의 검색을 통해 이루어지고 있다. 그리고 여기서 금전적 가치를 지니고 있다는 것은 제3자가 통신으로 전달되는 화폐를 가로채어 사용할 수 있다는 것을 말하며, 따라서 적절하게 암호화하여 교환하여야 한다. 본 시스템은 기존의 온라인 전자화폐를 사용한다고 가정하므로 전자화폐의 인출과정이나 화폐의 형태는 고려하지 않는다. 셋째, AMPS 서버는 신뢰성 있는 서버이며 고객들이 하나의 AMPS 서버에 집중되어 병목현상이 발생하는 것을 방지하기 위하여 다수의 AMPS 서버가 존재하는 것으로 가정한다.

2.3 프로토콜 설계

전체 시스템의 시나리오는 다음과 같다. 고객은 판매자의 사이트를 방문하면서 구매할 상품을 선택하고 선택된 상품의 정보와 판매자에게 지불할 전자화폐를 AMPS 서버에게 전달한다. AMPS 서버는 고객을 대신하여 판매자에게 상품 구매를 요청하여 판매자로부터 해당 상품과 상품 복호화 키를 받는다. 상품에 대한 기본적인 확인 작업을 한 후에 고객으로부터 받은 전자화폐를 발행기관에 제시하여 화폐와 금액을 확인한다. 발행기관으로부터 전자화폐가 유효하다는 회신을 받으면 AMPS 서버는 판매자에게 발행기관이 서명한 지불승인서를 전달하고 고객에게는 판매자로부터 받은 상품을 전자우편을 이용하여 전달한다. AMPS 서버가 고객을 대신하여 구매를 하게 되므로 고객이 직접 거래에 참여하지 않아도 된다. 따라서 고객의 무선 통신 메시지의 수와 고객에게 부과되는 계산량을 줄이는 효과를 얻을 수 있다. 또한 지불 프로토콜이 진행되는 동안 고객은 반드시 온라인일 필요가 없으므로 통신비용을 절감하는 효과도 있다.

본 시스템에서 전자화폐의 흐름만 요약하면 다음과 같다. 고객은 안전한 방법으로 전자화폐를 발행기관으로부터 인출하여 이동호스트에 보관하고 있다가 구매할 상품이 생기면 필요한 만큼의 전자화폐를 AMPS 서버로 전송한다. AMPS 서버는 수신한 화폐를 발행기관에게 전달하여 확인한다. 나중에 거래가 완료되면 판매자는 발행기관에 요청하여 자신의 계좌로 해당되는 금액을 이체 받는다. 특이한 점은 다른 시스템과 달리 판매자는 자신이 직접 고객으로부터 전자화폐를 받아 발행기관에 보내어 확인하고 정산 요청을 하지 않아도 된다. 본 시스템에서는 고객의 거래를 대행하여주는 AMPS 서버가 화폐의 확인까지 대신하여 주며, 전자화폐를 대신 발행기관에게 전달한다. 판매자는 나중에 지불승인서의 일련번호들을 모아 서명하여 발행기관에 전달하여 상품 대금을 받을 수 있다.

2.4 표기법

본 논문에서는 표 1에 제시된 표기법을 사용하여 제안된 지불시스템의 프로토콜을 기술하고 설명한다. 참고로 쉼표(,)는 구성 요소간에 결합(concatenation)을 나타내며, 아래 첨자는 키의 소유자 또는 난스(nonce)나 타임스탬프를 생성한 사용자를 나타낸다. 예를 들어 $+K_m$은 판매자 M의 공개키를 나타내며, K_{cm}은 고객 C와 판매자 M사이에 공유하기 위해 고객이 생성한 비밀키를 나타낸다.

2.5 AMPS 지불 프로토콜

AMPS 지불 프로토콜은 그림 2에 기술되어 있다. 지불 프로토콜에 대한 자세한 설명한 다음과 같다.

〔표 1〕 표기법

K	대칭형 알고리즘에서 사용되는 비밀키
+K	비대칭형 알고리즘에서 사용되는 공개키
-K	비대칭형 알고리즘에서 사용되는 개인키
{M}.K	암호키 K를 이용하여 메시지 M을 암호화한 블럭
N	난스(nonce)
T	타임스탬프
h(M)	메시지 M에 해시함수를 적용한 결과값
C	고객의 식별자, 회원 식별자, 고객의 전자우편주소
S	AMPS 서버
M	판매자의 식별자 또는 인터넷 주소
B	전자화폐 발행기관
coin	금전적 가치를 지니고 있는 전자화폐
pay-spec	지불명세서(M, goodID, price)
pay-cert	지불승인서(h(pay-spec), Nb(승인서 일련번호), Tb(승인서 발행시간))

메시지 1. (구매요청) C → S:	{NULL or C, N_c, goodID, price}.K_{cm}, {K_{cm}}.+K_m {M, C, N_c, {K_{cb}}.+K_b}. K_{cs},{M,h(pay-spec),coins}.K_{cb}, {K_{cs}}+K_s
메시지 2. (구매요청 전달) S → M:	{NULL or C, N_c, goodID, price}.K_{cm}, {K_{cm}}.+K_m
메시지 3. (상품전송) M → S:	{good}.K_{good}, {h({good}.K_{good}, N_c)}.-K_m, {{K_{good}}.K_{cm}}.+K_s {{h(pay-spec), price, N_m}.-K_m}.K_{mb}, {K_{mb}}+K_b
메시지 3'. (요청거절) M → S:	{reason, N_c}.-K_m
메시지 4. (화폐검증) S → B:	{M, h(pay-spec),coins}.K_{cb}, {K_{cb}}+K_b {{h(pay-spec), price, N_m}.-K_m}.K_{mb}, {K_{mb}}+K_b
메시지 5. (승인서 발급) B → S:	N_b, T_b, {N_m, h(pay-cert)}.-K_b
메시지 6. (승인서 전달) S → M:	N_b, T_b, {N_m, h(pay-cert)}.-K_b
메시지 7. (상품 전달) S → C:	{good}.K_{good}, h({good}.K_{good}, N_c), {K_{good}}.K_{cm}

〔그림 2〕 AMPS 지불 프로토콜

2.5.1 구매 요청

고객이 구매 요청을 하기 위해서는 먼저 판매자 사이트를 방문하여 구입할 상품을 선택하고 선택한 상품에 대한 정보를 얻어야 한다. 또한 고객은 프로토콜을 진행하기 앞서 AMPS 서버와 판매자의 인증서를 가지고 있어야 한다. 인증서는 지난 거래를 통해 이미 가지고 있는 상황일 수도 있고, 디렉토리 서버에게 요청하여 받을 수도 있다. 고객은 구매 요청을 하기 위해 두 개의 메시지를 구성하는데, 하나는 고객과 판매자만이 그 내용을 볼 수 있는 메시지이고, 다른 하나는 고객과 AMPS 서버만이 그 내용을 볼 수 있는 메시지이다. 고객은 이 두 메시지를 AMPS 서버에게 전달함으로써 구매 요청을 한다.

메시지 1의 첫 번째 암호문은 고객과 판매자만이 거래되는 상품과 상품 가격을 알 수 있도록 하기 위해서 임의의 대칭 암호키 K_{cm}을 생성하여 해당 고객이 요청한 상품임을 연결시켜 주는 난스 N_c와 상품코드, 그리고 가격을 암호화한다. 그리고 K_{cm}은 판매자의 공개키로 암호화하여 해당 판매자만이 복호화할 수 있도록 한다. AMPS 서버와 제3자는 판매자의 개인키를 모르기 때문에 고객이 어떤 상품을 구입하는지, 그리고 금액이 얼마인지를 알 수 없다. 즉 AMPS 서버는 고객을 대신하여 거래를 대행하여 주지만 어떤 상품을 어떤 가격에 구입하는지는 알 수 없고, 다만 어떤 판매자로부터 상품을 구입하는가만 알 수 있다. 고객은 할인이나 회원 서비스를 제공하는 판매자에게 상품을 구입할 경우에는 메시지에 고객의 회원 식별자를 추가할 수 있다. 만약 회원 식별자가 필요 없을 경우에는 판매자에게도 고객의 신원을 숨길 수 있다.

메시지 1의 두 번째 암호문은 AMPS 서버가 거래를 대행하기 위해 필요한 정보를 임의의 대칭 암호키 K_{cs}로 암호화한 메시지이다. 이 암호문은 구매 요청과 암호화된 상품을 연결하기 위한 난스 N_c, 상품을 수신하기 위한 고객 식별자 또는 전자우편주소, 접촉할 판매자를 가르쳐주기 위한 판매자 식별자, 그리고 발행기관의 공개키로 암호화되어 있는 K_{cb}로 구성되어 있다. {K_{cb}}.+K_b를 암호화하여 비밀성을 보장하는 이유는 판매자가 위의 메시지를 가로채어 상품을 전달하지 않고 지불승인서를 받을 수 없도록 하기 위함이다. AMPS 서버에게 전송되는 메시지 내에 전자화폐가 포함된 암호문은 판매자만이 전자화폐에 대한 권리를 주장할 수 있도록 하기 위해 판매자 식별자, 지불명세서의 해시값, 그리고 전자화폐를 K_{cb}로 암호화한다. 이것은 Ecash 시스템과 동일한 형태이다. 지불명세서는 고객이 구입할 상품에 대해서 판매자와 가격 협상한 결과이며 판매자, 상품코드, 그리고 상품가격으로 구성된다. 그리고 해시값을 통해 쉽게 내용을 예측할 수 없도록 판매자와 고객만이 알고 있는 랜덤한 난수를 추가할 수 있다.

메시지 1. (정산 요청) M → B:	{N_b, N'_b, ...}.K_{mb}, {h(N_b, N'_b, ...)}.-K_m, {K_{mb}}.+K_b
메시지 2. (정산요청 확인) B → M:	{h(N_b, N'_b, ...)}.-K_b

〔그림 3〕 정산 프로토콜

2.5.2 구매요청 전달 및 상품 전송

AMPS 서버는 고객으로부터 메시지 1을 받으면 자신의 공개키로 암호화된 메시지를 복호화하여 키를 얻은 다음 그 키로 암호화된 메시지를 복호화하여 거래해야 할 판매자를 확인하고 고객으로부터 받은 메시지를 판매자에게 전달한다. 판매자는 AMPS 서버로부터 메시지 2를 받으면 자신의 개인키를 이용하여 암호문을 복호화한다. 상품코드와 가격을 확인하고 요청한 가격에 상품을 판매할 수 있으면 상품 전송 메시지를 서버에게 전달한다.

판매자는 고객이 구매 요청한 디지털 상품을 임의의 대칭 암호키 K_{good}를 생성하여 암호화한다. 상품을 암호화한 이유는 제 3자가 통신 채널을 도청하여 상품을 얻을 수 없도록 하기 위한 것이다. 이 키는 고객이 생성한 세션키 K_{cm}과 서버의 공개키로 이중 암호화한다. 이중 암호화한 이유는 고객이 적은 금액 또는 불법적인 화폐를 사용할 경우에는 상품을 받아볼 수 없도록 하기 위한 것이다. 판매자는 전송한 상품의 무결성과 암호화된 상품이 고객이 요청한 상품임을 확인할 수 있도록 고객이 제시한 난스와 상품의 해시값을 만든 후 이것을 서명하여 전달한다. 판매자의 서명이 되어 있고 최근성을 보장하는 난스가 포함되므로 엉뚱한 상품을 전달할 경우에는 나중에 누가 부정을 했는지 밝혀낼 수 있다. 가격과 지불명세서를 서명한 것을 암호화하여 AMPS 서버가 볼 수 없도록 한 것은 거래의 세부내용을 서버가 모르게 하기 위함이다. 여기서 N_m은 판매자가 지불승인서를 식별하기 위한 난스이다. 판매자는 메시지 2를 받은 후에 상품 판매를 거절할 수 있다. 거절을 할 경우에는 메시지 3'를 AMPS 서버에게 전달한다.

2.5.3 지불처리 및 상품 배달

AMPS 서버는 지불 프로토콜의 원자성을 보장하기 위해 판매자에게 상품 대금을 지불하기 전에 고객으로부터 수신한 전자화폐의 유효성을 발행기관으로부터 검증 받는다. AMPS 서버는 고객으로부터 받은 전자화폐가 포함된 암호문과 함께 판매자가 상품 전송시 제시한 암호문을 발행기관에 전송한다. 발행기관은 첫 번째 암호문을 복호화하여 판매자 식별자에 해당하는 공개키를 디렉토리 서버 또는 자신의 캐시에서 얻어 고객이 제시한 지불 명세서의 해시값과 판매자가 제시한 지불명세서의 해시값이 일치하는지 비교한다. 또한 전자화폐의 유효성, 이중사용 여부를 확인하고, 전자화폐의 금액이 판매자가 제시한 상품 가격과 일치하는지 확인한다. 전자화폐의 유효성 확인과 이중사용에 대한 검증은 사용하는 기존 화폐에서 사용하는 방식으로 이루어진다. 예를 들어 Ecash 화폐를 사용하면 화폐에 있는 은행의 서명을 확인하고, 입금 데이터베이스를 검사하여 이미 입금된 화폐인지를 확인할 것이다. 확인이 되면 지불승인서를 발급하여 AMPS 서버에게 전달한다.

지불승인서는 판매자가 제시한 지불명세서와 승인서의 일련번호, 승인시간으로 구성되며, 지불승인서의 해시값과 판매자가 지불승인서를 식별할 수 있도록 판매자가 제시한 난스를 서명하여 전달한다. 또한 해시값을 확인할 수 있도록 승인서의 일련번호, 승인시간을 서명과 함께 전달한다. AMPS 서버는 고객이 제시한 전자화폐가 유효하다는 지불승인서를 발행기관으로부터 받으면 승인서를 확인하고 판매자에게 그대로 전달해준다. 이와 더불어 고객에게 판매자로부터 받은 상품을 메시지 7과 같은 형태로 전자우편을 이용하여 전달한다. 판매자는 AMPS 서버로부터 받은 지불승인서들을 보관하고 있다가 나중에 지불승인서의 일련번호들을 그림 3과 같이 모두 발행기관에게 보내 지불 정산을 요청한다.

2.6 상태 질의 프로토콜

본 시스템의 경우 고객이 구매 요청 메시지를 전송하기 전이나 그리고 판매자가 상품을 전송하기 전에는 거래를 취소할 수 있으나 그 이후에는 거래 자체를 취소할 수 없다. 상태 질의 프로토콜은 고객 또는 판매자가 진행중인 프로토콜의 상태를 문의하기 위해 사용하는 프로토콜로서 각 처리단계에서 통신이 단절되는 등 장애가 발생할 수 있는 상황에서 원자성을 보장하기 위해 필요하다. 고객은 구매 요청과 함께 전자화폐를 전송하므로 만약 그 이후에 거래에 문제가 발생하면 금전적 손해를 볼 수 있으므로 상태를 확인하고 필요한 조치를 취할 수 있어야 한다. 판매자는 상품과 상품을 암호화한 키를 같이 전송하므로 만약 지불승인서를 받지 못할 경우에는 마찬가지로 금전적 손해를 볼 수 있다. 고객은 상품 구매 요청 메시지를 전송한 후에 구입한 상품이 오지 않을 경우, 판매자는 지불승인서를 받지 못하였을 경우 각각 그림 4에 기술되어 있는 (a)와 (b) 프로토콜을 수행한다.

그림 4에 제시된 상태 질의 프로토콜을 처리하기 위해 서버가 트랜잭션마다 영구적으로 관리하여야 하는 정보에는 트랜잭션 고유번호(N_c), 고객(C), 판매자(M),

메시지 1. (상태 질의) C → S:	$\{C, N_c\}.K_{cs}$, $\{K_{cs}\}.+K_s$
메시지 2. (질의 응답) S → C:	$\{reason, N_c\}.K_{cs}$ or Message 7

(a) 고객 상태질의 프로토콜

메시지 1. (상태 질의) M → S:	$\{M, N_c\}.K_{ms}$, $\{K_{ms}\}.+K_s$
메시지 2. (질의 응답) S → M:	$\{reason, N_c\}.K_{ms}$ or Message 6

(b) 판매자 상태질의 프로로콜

〔그림 4〕 AMPS 상태질의 프로토콜

트랜잭션 상태, 요청시간(T_s), $\{good\}.K_{good}$, $\{K_{good}\}.K_{cm}$,, $\{h(\{good\}.K_{good}, N_c)\}.-K_m$, 지불승인서 정보($N_b$, T_b, $\{N_m, h(pay\text{-}cert)\}.-K_b$) 등이 있고, 일시적으로 보관하여야 하는 정보로는 고객이 지불한 전자화폐 등이 있다. 그러나 보관하는 정보 가운데 암호화된 디지털 상품은 그 크기가 매우 클 수 있으므로 서버의 저장 공간 측면에서 문제가 될 수도 있다. 그러나 이런 문제는 다음과 같은 몇 가지 방법을 사용하여 극복이 가능하다. 첫째, 디지털 상품은 일정한 기간 동안만 보관하고, 그 기간이 지나면 무조건 삭제하는 방법이다. 만약 그 이후에 문제가 발생하면 판매자에게 사유를 밝히고 다시 요청한다. 둘째, 고객으로부터 확인메시지를 추가로 요구하여 확인 메시지를 받을 때까지만 보관하는 방법을 사용할 수도 있다. 이와 같이 본 시스템은 거래 중에 발생한 여러 가지 상황을 대비하여 처리 결과를 조회할 수 있는 프로토콜을 제공함으로써 원자성을 보장한다.

Ⅲ. 성능 분석

3.1 안전성 분석

본 장에서는 제안된 지불시스템의 안전성과 원자성을 분석하기 위해서 지불시스템에서 발생할 수 있는 여러 가지 상황을 제시하고 그런 상황에서 안전하다는 것을 보여준다.

3.1.1 고객의 부정행위

고객이 부정을 행하는 경우를 고려하면 첫째 유효하지 않은 전자화폐를 사용할 경우와 둘째 상품을 받고서 요청한 상품과 다르다고 할 경우를 들 수 있다. 첫 번째의 경우는 AMPS 서버가 고객에게 상품을 전달하기 앞서 고객이 제시한 전자화폐의 유효성을 발행기관에 의뢰하므로 유효하지 않은 전자화폐를 사용한 고객은 상품을 받을 수 없게 된다. 또한 상품에 해당하는 금액보다 적은 금액을 전송하여도 발행기관이

메시지 1. C → S: $\{req\text{-}code, C, N_c, K_{cm}\}.K_{cs}$, $\{K_{cs}\}.+K_s$

〔그림 5〕 분쟁해결 프로토콜

판매자가 제시한 금액과 비교하기 때문에 이 경우에도 고객은 상품을 받을 수 없게 된다. 두 번째의 경우 고객은 AMPS 서버에게 요청한 상품이 아니라고 이의를 신청할 수 있다. 이 분쟁을 해결하기 위해서 AMPS 서버는 고객으로부터 상품 구매 요청시 사용하였던 키를 요청한다. 실제 사용하는 프로토콜은 그림 5에 기술되어 있다.

AMPS 서버는 보관된 메시지 중에 N_c에 해당하는 상품 요청 메시지를 분쟁해결 프로토콜의 메시지 1에 있는 키를 이용하여 복호화 하여 고객이 요청한 상품 코드와 상품을 확인한다. 만약 상품이 해당 고객의 요청한 상품과 다를 경우, 또는 키 문제로 복호화가 올바르게 이루어지지 않으면 판매자에게 이 사실을 알리고 배상을 요구한다. 판매자는 상품 전송시 개인키로 암호화된 상품의 해시값을 서명하였기 때문에 부인할 수 없다. 반대로 올바른 상품임이 밝혀지면 고객의 부정행위임이 들어나게 된다.

3.1.2 판매자의 부정행위

판매자가 부정을 행하는 경우는 고객이 요청한 상품보다 값이 싼 다른 상품을 제공하는 경우와 상품을 볼 수 있는 키를 올바르게 전송하지 않는 경우를 들 수 있다. 전자의 경우는 고객이 AMPS 서버에게 요청한 상품과 다르다고 이의를 신청하면 판매자의 부정이 밝혀진다. 후자의 경우에도 전자와 마찬가지로 고객이 AMPS 서버에게 이의를 신청하면 판매자의 부정을 쉽게 밝힐 수 있다. 또한 두 경우 모두 판매자가 서명한 상품의 해시값이 보관되어 있기 때문에 판매자는 부인할 수 없다. 그러나 실제 판매자는 이런 부정행위를 하지 않을 것이다. 왜냐하면 판매자가 얻을 수 있는 이익이 소액이며 부정행위가 드러나면 신용도가 떨어져 고객들로부터 외면 당할 위험이 있기 때문이다.

3.1.3 제 3자의 공격

제 3자가 본 시스템을 공격한다면 크게 두 가지 목표를 가지고 할 수 있다. 하나는 통신 채널로 전달되는 상품을 가로채어 불법으로 열람하는 것이고 다른 하나는 전자화폐를 가로채어 사용하는 것이다. 상품을 가로채어 열람하기 위해서는 암호화된 상품을 복호화하는 키를 얻어야 한다. 만약 프로토콜이 정상적으로 진행되었다면 고객이 생성한 세션키를 모르기 때문에 상품 복호화 키를 얻을 수 없다. 따라서 제 3자는 프로토콜에 능동적으로 공격하여 얻을 수밖에 없다. 가능한 능동 공격을 살펴보면 제 3자는 고객이 AMPS 서버에게 전송하는 암호문 중 판매자에게 전달될 부분 또는 서버가 판매자에게 중계하는 구매 요청 메시지를 자신의 메시지로 바꾸어 공격하는 방법을 생각할 수 있다. 판매자는 상품 전송 메시지를 만들어 전달하지만 AMPS 서버의 개인키가 없을 경우에는 상품 복호화 키를 얻을 수 없다. 또한 고객이 선택한 난스와 공격자가 선택한 난스가 다를 경우에는 AMPS 서버는 판매자가 서명한 암호화된 상품의 해시값을 확인하여 프로토콜 진행에 문제가 있음을 알게 된다. 만약 동일한 난스를 선택할 수 있으면 정상적으로 프로토콜이 진행되며 공격자는 전자우편으로 전달되는 상품을 가로채어 상품을 열람할 수 있다. 따라서 난스는 예측할 수 없는 것으로 선택되어야 한다.

공격자가 전자화폐를 가로채어 자신의 이득을 위해 사용하기 위해서는 고객이 AMPS 서버에게 전송하는 메시지에 암호화되어 있는 전자화폐를 복호화하거나 AMPS 서버만 사용할 수 있도록 암호화한 $\{K_{cb}\}_{+K_b}$를 얻어야 한다. 그러나 AMPS 서버의 개인키를 모르기 때문에 불가능하다. 공격자가 메시지 4를 발행기관에 도달하지 못하도록 가로채는 공격을 생각할 수도 있다. 그러나 AMPS 서버가 발행기관으로부터 확인 메시지를 기다리기 때문에 그 사이에 공격자가 이득을 취하기는 거의 불가능하다.

3.2 제안한 지불시스템의 특성 분석

소액 상품의 거래를 위해 제시된 지불시스템은 현재 수십 종류에 이르고 있고 각 시스템별로 독특한 특징을 가지고 있다. 그러나 아직까지 이상적인 시스템이 개발된 것은 아니고 각 시스템별로 상호 장단점을 가지고 있다. 이 절에서는 구매자의 익명성, 구매 상품의 비밀성, 보증 배달 원자성에 대해 본 지불시스템과 기존의 지불시스템과 비교하고, 이동컴퓨팅 환경이므로 이동호스트를 사용할 고객의 공개키 연산 횟수를 다른 시스템과 비교한다. 표 2는 기존의 지불시스템과 본 지불시스템이 지불시스템의 요구조건에 대해서 어떤 부분을 만족시키고 있는지를 비교한 결과이다.

〔표 2〕 지불시스템 특성 비교

요구조건 / 시스템	구매자의 익명성	구매상품의 비밀성	원자성
Ecash	○	×	×
NetCash	×	×	×
PayMe	×	×	×
Millicent	×	×	×
NetCent	○	×	○
NetBill	×	○	○
AMPS	△	○	○

3.2.1 익명성과 상품의 비밀성

본 논문에서 제시한 지불시스템은 기존의 개발된 전자화폐를 사용한다고 가정하고 있으므로 사용자에 대한 익명성은 사용하는 전자화폐에 의존한다. 그러나 기존의 지불시스템은 고객과 판매자간의 거래이기 때문에 판매자는 고객의 정보를 알 수 있다. 본 지불시스템에서는 AMPS 서버가 고객을 대신하여 구매하도록 하고 있어 고객을 판매자로부터 숨길 수 있다. 또한 고객과 판매자 사이에 전송되는 상품과 상품정보를 AMPS 서버와 제 3자에게 숨김으로서 고객의 프라이버시를 보호하고 있다. 그러나 분쟁을 해결하여야 하는 경우에는 거래정보가 AMPS 서버에게 노출되는 문제가 있다.

3.2.2 원자성

이동컴퓨팅 환경에서는 지불 트랜잭션의 원자성을 보장하는 것이 매우 중요하다. 본 지불시스템은 AMPS 서버를 이용하여 고객이 상품 구매 대금을 올바르게 지불하지 않으면 상품을 받을 수 없도록 하였고, 판매자가 상품을 전송하지 않으면 지불 대금을 받을 수 없도록 하여 원자성을 보장하고 있다. 또한 상태 질의 프로토콜과 분쟁 해결 프로토콜을 제공하여 고객과 판매자 모두 피해를 보는 경우가 발생하지 않도록 보장하고 있다.

판매자와 고객간의 분쟁이 발생하였을 경우 기존의 지불시스템은 일반적으로 고객과 판매자가 제시하는 증거들을 사용하여 부정 행위자를 판별한다. 본 지불시스템도 고객이 제시한 증거를 이용하여 AMPS 서

버가 분쟁을 해결하여 준다. 본 시스템에서 발생할 수 있는 분쟁은 고객이 요청한 상품과는 다른 상품을 고객이 받았을 경우에만 있고 다른 분쟁은 사전에 발생하지 않도록 프로토콜 자체에서 보장하고 있다. 따라서 판매자가 분쟁 해결을 요청하는 경우는 본 시스템에서는 일어나지 않는다. 분쟁을 해결하는데 있어서도 트랜잭션 정보를 AMPS 서버가 보유하고 있기 때문에 쉽게 할 수 있다. 다만 원자성을 보장하기 위해 AMPS 서버가 트랜잭션마다 유지하고 관리하여야 하는 정보가 많은 단점이 있다. 특히 암호화된 상품을 보관하고 있어야 하는데 만약 디지털 상품이 멀티미디어 파일일 경우에는 그 크기가 문제가 될 수 있다. 이 문제는 2장에서 언급한 것처럼 일정기간만 보관한 후에 삭제하는 방법을 사용하거나 고객으로부터 확인 메시지를 추가로 요구하는 방법을 사용하여 해결할 수 있다. 끝으로 AMPS는 전자화폐의 인출과정을 고려하지 않으므로 인출 원자성은 고려하고 있지 않으나 예금 원자성은 보장한다.

3.2.3 이중 사용방지

기존의 개발된 대부분의 전자화폐들은 이중 사용방지를 위해서 사용된 화폐의 일련번호를 데이터베이스에 기록하며, 지불을 예치할 때 이 데이터베이스를 검색하여 이중사용 여부를 판별한다. 본 지불시스템에서도 전자화폐를 수용하는 발행기관에서 이중사용 여부를 확인하여야 하는 전자화폐를 사용한다고 가정하고 있다. 그러나 다른 지불시스템과는 달리 판매자는 전자화폐를 고객으로부터 직접 수신하지 않으며, 전자화폐를 직접 발행기관에게 전달하여 지불 정산을 요청하지 않는다. 본 시스템에서는 AMPS 서버가 대신 고객으로부터 받은 화폐를 발행기관에 전달하여 주며 이 과정에서 나중에 정산을 할 수 있는 지불승인서를 받아 판매자에게 중계하여 준다. 따라서 판매자는 기존 시스템에 비해 보다 편리하게 정산을 요청할 수 있다. 그러나 본 시스템은 이와 같은 가정 때문에 온라인으로 지불할 수밖에 없는 단점이 있다. 따라서 기존에 제시된 오프라인 지불이 가능한 전자화폐를 본 시스템에 도입하여 확장하거나 본 시스템의 장점을 그대로 유지할 수 있는 오프라인 지불이 가능한 새로운 화폐를 개발하는 연구가 필요하다.

3.2.4 공개키 알고리즘 연산 횟수 및 암호화하는 메시지 크기 비교

이동컴퓨팅 환경에서는 이동 호스트의 특성 때문에 공개키 암호 연산을 적게 사용할 수록 효율적이다. 기존 시스템들은 이런 특성을 고려하지 않고 있기 때문에 이동 호스트에 많은 연산 부담을 줄 수도 있다. 또한 소액 지불시스템을 지원하기 위해서는 지불 대금보다 지불 메커니즘의 비용이 높지 않도록 지불과정이 매우 효율적이어야 하는데 공개키 암호 연산은 이와 같은 측면에서도 적합하지 않다. AMPS는 새로운 전자화폐를 제시하지 않고 기존 화폐를 사용한다고 가정하고 있기 때문에 인출 과정에서 소요되는 공개키 연산은 본 비교에서 고려하지 않으며, 인증서를 확인하기 위한 연산도 모두 공통적으로 필요하므로 고려하지 않는다.

〔표 3〕 공개키 연산 횟수 및 암호화되는 메시지 크기 비교

요구조건 / 시스템	공개키 연산 횟수		메시지 크기	
	암호화	복호화	암호화	복호화
Ecash	1	1	128	영수증
NetCash	1	1	320 + 전자화폐	영수증
PayMe	1	1	224 + 전자화폐	영수증
NetCent	1	0	224	0
NetBill	3	1	448	영수증
AMPS	3	0	384	0

AMPS에서 고객은 전자화폐의 도난을 방지하고, 고객과 판매자 사이의 거래 내역의 비밀성을 보장하기 위해 세 번의 공개키 암호 연산을 사용하고 있다. 지불 프로토콜의 메시지 1를 보면 고객은 세 개의 세션키를 각각 암호화하기 위해 세 번의 공개키 암호 연산을 사용하였다. 다른 시스템의 공개키 암호 연산의 사용 현황을 보면 다음과 같다. 온라인 Ecash 시스템에서 고객은 전자화폐를 판매자에게 지불하는 과정에서 세션키를 암호화하기 위해 그리고 판매자로부터 받은 영수증을 확인하기 위해 두 번의 공개키 연산을 사용한다. NetCash 시스템에서 고객은 지불하기 위해 전자화폐, 두 개의 대칭키, 그리고 상품코드를 판매자의 공개키로 암호화하여 전송하며, 나중에 영수증을 확인하기 위해 공개키 암호 연산을 사용하여야 한다.[6] PayMe 시스템은 NetCash와 같은 횟수의 공개키 알고리즘을 사용한다.[7] Netcent는 전자지불지시서를 서명하기 위해 한 번의 공개키 연산을 사용한다.[8] NetBill은 Kerberos 티켓을 수신하기 위해 두 번의 공개키 연산이 필요하며,[9] 지불 단계에서 전자지불지시서를 서명하고 NetBill 서버가 발급한 영수증을 확인하기

위해 두 번의 공개키 연산이 필요하다.

표 3은 고객이 연산해야 하는 공개키 연산 횟수와 고객이 공개키 알고리즘을 사용하여 암호화하는 메시지의 크기를 비교한 것이다. 공개키로 암호화되는 데이터의 크기는 시스템마다 다를 수 있으므로 대칭키의 길이는 128비트, 메시지 해시값의 길이는 160비트, 참여자 식별자와 난스는 64비트, 시간 정보, 금액, 기타 수치 데이터는 32 비트라고 가정하여 크기를 비교하였다. 표 3에서 보는 바와 같이 공개키 연산횟수는 기존 시스템과 크게 두드러진 향상은 없다. 그러나 기존 지불시스템들은 상품의 비밀성과 원자성을 제공하지 않는 반면 AMPS는 상품의 비밀성과 원자성을 보장하고 있다. 표 2에서도 알 수 있듯이 NetBill, Netcent를 제외한 다른 시스템들은 대부분 원자성을 보장하지 못하고 있으며 NetBill을 제외한 다른 시스템들은 상품의 비밀성을 고려하고 있지 않다. 다른 시스템이 본 시스템이 제공하는 원자성과 디지털 상품의 비밀성을 보장하도록 보완한다면 공개키 연산이 늘어 날 것이다. 따라서 AMPS는 기존 시스템들보다 적은 공개키 연산을 사용한다고 말할 수 있다.

Ⅳ. 결 론

오늘날 전자상거래 시스템에서 가장 중요한 요소인 안전한 지불시스템을 확보하려는 선진국들의 경쟁은 이미 시작되었다. 곧 상거래의 대부분이 가상 공간에서 이루어지게 될 것이며 이를 위한 시스템들이 현재 세계의 곳곳에서 개발되고 있다. 이 논문에서는 현재 주목받고 있는 전자화폐, 전자지불시스템, 그리고 이동컴퓨팅 환경에서 요구되는 조건들을 분석하여 이동컴퓨팅 환경에서 디지털 상품을 구매하기 위한 새로운 소액지불시스템 AMPS를 제안하였다.

본 지불시스템은 이동컴퓨팅 환경이라는 상황에서 효율적인 지불이 가능하도록 중개자의 역할을 하는 신뢰성 있는 AMPS 서버를 두어 고객을 대신하여 거래를 유선망에서 대행하도록 하고 있다. 고객은 AMPS 서버에게 구매하고자 하는 상품 정보와 전자화폐를 전달하면 실제 거래와 지불에 필요한 여러 가지 확인 작업은 서버가 대신하여 준다. 모든 확인 작업이 끝나면 서버는 판매자로부터 받은 상품을 고객에게 전자우편을 이용하여 보내준다. 이 기간동안에 고객은 연결을 유지하고 있을 필요가 없어 이동 호스트인 고객 시스템의 통신 비용과 계산량을 감소시키는 효과를 얻을 수 있다. 또한 고객과 판매자 사이에 전송되는 상품과 상품정보를 AMPS 서버와 제 3자에게 숨김으로써 고객의 익명성과 상품의 비밀성을 보장하고 있을 뿐만 아니라 판매자에게 고객의 정보를 숨길 수도 있다.

이 시스템의 또 다른 특징은 오류율과 단절이 높은 이동컴퓨팅 환경에서 고객과 판매자 어느 쪽도 피해를 입지 않도록 하기 위해서 원자성을 보장한다는 점이다. 이를 위해 판매자는 상품을 암호화하여 전송하고 AMPS 서버는 판매자에게 지불하기 전에 고객이 제시한 전자화폐의 유효성을 검증한다. 그리고 분쟁이 발생하였을 때 이를 해결할 수 있도록 AMPS 서버는 필요한 거래 내역을 보관한다.

그러나 본 지불시스템은 이중사용 여부를 발행기관으로부터 확인하여야 하는 절차 때문에 현재는 온라인으로 지불할 수밖에 없다. 따라서 시스템이 지니고 있는 현재의 장점을 수용하면서 오프라인으로 지불할 수 있는 새로운 전자화폐를 개발하거나 오프라인 지불이 가능한 화폐를 본 시스템에 이식하여 확장하는 연구가 필요하다. NetBill은 Kerberos 티켓을 이용하여, Millicent는 판매자가 화폐를 발행하도록 하여 동일한 사이트에 반복 구매를 할 때 편의를 도모하여 주는 기능이 있으나 본 시스템은 이와 같은 기능을 제공하지 못하고 있다[10,11]. 따라서 소액지불시스템에 보다 적합하도록 이런 기능을 추가하는 연구도 필요하다.

참 고 문 헌

[1] David Chaum, "Blind Signature for Untraceable Payments," *Advances in Cryptology, Proc. of Crypto'82*, 1982.

[2] M. Satyanarayanan, "Fundamemtal Challenges in Mobile Computing," *Proc. of the 15th ACM Symp. on Principle of Distributed Computing*, pp. 1-7, May 1996.

[3] J. D. Tygar, "Atomicity in Electronic Commerce," *Proc. of the 15th ACM Symp. on Principles of Distributed Computing*, pp. 8-26, May 1996.

[4] Shouhuai Xu, Moti Yung, Gendu Zhang, and Hong Zhu, "Money Conservation via Atomicity in Fair Off-Line E-Cash," *Proc. of the 2nd Int. Information Security Workshop*, LNCS 1729, pp. 14-31, 1999.

[5] B. Schoenmakers, "Basic Security of the ecash Payment System," State of the Art in Applied

Cryptography, Courses on Computer Security and Industrial Cryptography, LNCS 1528, June 1997.

[6] R. Rivest and A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes," *Proc. of 1996 Int. Workshop on Security Protocols*, LNCS 1189, pp. 69-87, 1996.

[7] G. Medvinsky and B. C. Neuman, "NetCash: A design for practical electronic currency on the Internet," *Proc. of the 1st ACM Conf. on Computer and Communications Security*, Nov. 1993.

[8] M. Peirce, "PayMe: Secure Payment for World Wide Web Services," B. A. (Mod) Project Report, Computer Sci-ence Department, Trinity College Dublin, May 1995.

[9] Tomi Poutanen, heather Hinton and Michael Stumm, "NetCents: A Light-weight Protocol for Secure Micro-payments," *Proc. of the 3rd USENIX Workshop on Electronic Commerce*, September 1998.

[10] Benjamin Cox, J. D. Tygar, and Marvin Sirbu, "Netbill Security and Transaction Protocol," *Proc. of the 1st USENIX Workshop on Electronic Commerce*, pp. 1-12, July 1995.

[11] M. S. Manasse, "The Millicent protocol for electronic commerce," *Proc. of the 1st USENIX Workshop on Electronic Commerce*, July 1995.

〈著者紹介〉

김 상 진 (Sangjin Kim) 정회원
1995년 2월 : 한양대학교 전자계산학과(학사)
1997년 2월 : 한양대학교 전자계산학과(석사)
1997년 3월~현재 : 한양대학교 전자계산학과(박사과정)
〈관심분야〉 암호기술 응용, 전자지불시스템, 이동컴퓨팅

성 주 환 (Juhwan Sung)
1998년 2월 : 한양대학교 전자계산학과(학사)
2000년 2월 : 한양대학교 전자계산학과(석사)
2000년 3월~현재 : (주)동성정보통신 연구원
〈관심분야〉 암호기술 응용, 전자상거래, 이동컴퓨팅

오 희 국 (Heekuck Oh) 종신회원
1983년 : 한양대학교 전자공학과
1989년 : 아이오아주립대학 전자계산학과(석사)
1992년 : 아이오아주립대학 전자계산학과(박사)
1993년~1994년 : 한국전자통신연구원 선임연구원
1995년~현재 : 한양대학교 전자컴퓨터공학부 조교수
〈관심분야〉 암호이론, 전자상거래, 이동컴퓨팅