인터넷상에서 소액지불을 위한 전자지불 프로토콜 설계및 구현

김진하, 김태형,류재철
충남 대학교 컴퓨터 과학과

(A Design and Implementation of Electronic Payment Protocol for Micropayment On the Internet)

Kim Jin Ha, Ryou Jae Cheol
Chungnam National University Computer Science Dept.

1. 서론

인터넷 사용의 계속적이고 폭발적인 증가에 따라 많은 사람들이 정보를 얻는데 인터넷에 의존하게 되었으며 그에 따라 인터넷은 단순히 정보를 얻는 공간뿐만이 아니라 기업이나 상인들에게 많은 소비자들을 접촉할 수 있는 중요한 장소가 되고 있다. 현재 World Wide Web 은 대부분이 기업이나 개인의 홍보용으로 사용하고 있지만 그와 동시에 많은 사람들이 인터넷을 통한 구매와 판매를 꾸준히 시도하고 있으며 현재 대부분 무료로 제공되는 정보 서비스가 점차 유료화됨에 따라 인터넷상에서의 상거래를 위한 지불 메카니즘이 필요하게 되었다.

이러한 시점에서 많은 기업이나 단체는 가까운 미래에 불어닥칠 인터넷을 통한 전자상거래를 위해 많은 프로토콜과 제품을 발표하고 있으며 그중에서 몇가지는 이미 시험사용 단계까지 와있다. 현재 제시된 전자 상거래 프로토콜로써는 크게 크레디트 카드를 이용하는 방식, 전자수표 방식, 전자화폐 방식으로 나뉘며 이 세가지의 프로토콜은 각각의 장단점을 가지고 인터넷상에서 공존하고 있는 실정이지만 많은 사람들이 전자화폐를 가장 이상적이며 궁극적인 전자 상거래의 구현으로 꼽고 있다.

그러나 현재 많은 프로토콜이 개발되었음에도 불구하고 이러한 프로토콜을 인터넷상에서의 잡지, 신문, 참고서적 구독 등의 온라인 서비스에 적용시키기에는 어려움이 따른다. 전자 상거래를 하기 위해서는 금융기관과의 연계나 브로커의 서비스 등에 필요한 비용이 드는데 일반적으로 온라인 서비스는 가격이 너무 낮아서 이러한 비용을 감당할 수 없다. 따라서 이러한 저가의 유료 온라인 서비스를 위해서 보안의 정도를 낮추고 전자 상거래에 드는 비용을 낮추는데 중점을 둔 NetBill, Millicent, PayWord, MicroMint 등의 몇가지 프로토콜이 제안되어 있다. NetBill 은 다양한 서버가 존재하여 여러 서비스(예를 들면 익명성을 제

공하기 위한 의사명을 제공하는 서비스)를 제공하는 전자수표 방식의 프로토콜이나 다른 프로토콜에 비해서 전자 서명이 많이 쓰여 비용과 시간이 많이 드는 단점이 있다.[2] Millicent 는 해쉬함수를 이용해서 계산 비용을 줄였으며 거래를 원하는 사용자는 브로커를 통해서 상인의 스크립을 구매해서 거래를 시작할 수 있다. 그러나 사용자가 새로운 상인과 거래를 할 때마다 브로커와 온라인으로 통신을 통해 상인의 스크립을 구입해야 하므로 브로커의 부하가 증가한다.[2] PayWord 는 payword 체인을 이용하여 전자서명을 줄이고 오프라인으로 결재를 할 수 있도록 하여서 서버와의 병목현상을 감소시킨 사용자의 예금계좌나 크레디트 방식을 기반으로 제안된 소액지불 프로토콜이다.[3]

PayWord 지불 프로토콜은 다른 여타의 프로토콜에 비해 참여 개체간의 통신의 양과 전자서명을 현저히 줄여 소액지불에 바람직한 특성을 지녔다. 본 논문에서 제안하는 프로토콜은 인터넷상에서 웹페이지등의 온라인으로 제공되는 소액 지불을 위한 프로토콜로써 전자화폐 방식을 기본으로 PayWord 의 소액지불에 바람직한 특성을 수용하여 거래에 드는 비용을 줄이고 전자화폐의 장점을 살리도록 한 방식이다.

기존의 전자 화폐가 화폐를 받았을 때 마다 온라인으로 서버에 연결하여 유용성을 검증한 것과는 달리, 거래를 시작할 때 화폐에 대한 인증서(Commitment)만을 온라인으로 검사한다. 그 후의 계속적인 거래는 일정한 금액을 가진 토큰을 통해서 오프라인으로 거래가 이루어 지며 상인은 어느 특정한 시간에 그동안 사용자에게 받은 토큰을 모두 브로커에게 보내 한번에 정산한다.

본 논문의 2 장에서는 관련연구로써, 제안한 프로토콜에 많은 영향을 끼친 전자화폐 방식의 NetCash 와 소액지불 방식인 PayWord 에 대해 기술되어 있으며, 3 장에서는 본 논문을 통해서 제안하는 프로토콜의 방식과 장단점에 대해서, 그리고 마지막 4 장에서는 결론과 향후 연구방향이 기술되어 있다.

2. 관련 연구

2.1 Netcash

NetCash 는 여러 도메인에서 사용할 수 있는 익명성을 보장하는 전자화폐이다. [4] NetCash 는 분산되어 있는 화폐발행서버(Currency server)로 이루어져 있으며 화폐발행서버는 익명성을 보장하는 화폐와 비익명성인, 예를 들어 전자 수표(NetCheque)등의 금융도구와의 교환 서비스를 제공한다. 화폐발행서버는 전자화폐의 발행을 위해 중앙은행같은 기관의 인증이 필요하므로 NetCash 프로토콜에서는 이러한 중앙 인증기관을 FIC(Federal insurance

corporation)이라 한다. <그림 1>은 화폐발행서버의 인증서의 내용이며 모든 내용은 FIC 의 비밀키(K_{FIC}^{-1})로 전자서명이 이루어 진다.

{Certif_id, CS_name, Kcs, issue_date, exp_date}K_{FIC}^{-1}

Certif_id : 화폐발행서버의 ID
CS_name : 화폐발행서버의 이름
Kcs :화폐발행서버의 공개키
issue_date : 인증서를 발급한 날짜
exp_date : 인증서의 만기일

<그림 1> 화폐발행서버의 인증서

<그림 2>는 전자화폐의 모습이다. 전자화폐는 화폐발행서버의 비밀키로 암호화된 일련의 번호(serial number)를 가지고 있으며 이 일련의 번호는 화폐가 발행될 때 갖게 되는 정보로써 서버는 이 번호를 저장하여 화폐의 유용성을 저장하므로 전체적으로 유일해야 한다. 사용자는 위의 화폐를 이용하기 위해서 자신이 등록한 서버에 화폐를 신청해야 한다. 화폐발행서버는 화폐발행 신청요구를 받으면 사용자를 확인하고 화폐를 발행해 준다. 전자화폐 거래시 화폐가 서버에게 보내지면 서버는 데이타 베이스에 그 번호가 존재하는지를 검사하여 만약 번호가 존재하면 그 화폐는 유용한 것이므로 새로운 화폐를 상인에게 발행해주고 화폐의 번호를 데이타베이스에서 삭제한다. 그러므로 번호가 발견되지 않는 것은 이중사용을 하는 것이므로 교환을 중지한다.

{CS_name, s_addr, exp_date, serial_num, coin_val}Kcs^{-1}

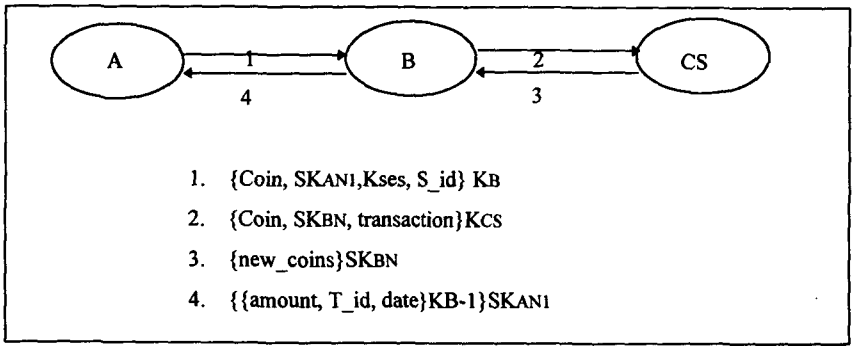
CS_name : 화폐발행서버의 이름.
s_addr : 화폐발행서버의 주소
exp_date : 화폐의 유효날짜
coin_val : 화폐의 금액

<그림 2> 전자화폐

1) Netcash 교환 프로토콜

거래에 참여하는 개체는 사용자(Client), 상인(Merchant), 화폐발행서버(Currency Server)

이다. 사용자나 상인은 A 또는 B 로 표시되며 서버는 CS 로 표기한다. X 는 모든 파트를 표시할 수 있다. 암호화된 내용은 {} 로 표시하며 $Kkey^{-1}$은 비밀키를 의미하고 공개키는 Kkey 로 나타낸다. SK 는 공용키를 말한다.



<그림 3> 이중사용 방지를 위한 프로토콜

<그림 3>은 상인과 거래를 하는 전형적인 모습이다. 1 단계에서 A 는 B 에게 코인, A 의 임시키(SKAN1), 세션키(Kses), 화폐발행서버의 *ID(S_id)*를 B 의 공개키로 암호화하여 보낸다. 세션키는 A 가 후에 B 에게 계속적인 서비스를 받게 될 때 B 에게 제출해야 하는 정보이다. 2 단계에서 B 는 코인의 유용성을 검증하기 위하여 서버에게 코인과 B 의 임시키(SKBN), Transaction (NetCash 는 NetCheque 와 연동하는 프로토콜이므로 서버에게 코인을 보낼 때 새롭게 발급받는 것을 코인뿐만 아니라 수표로써도 할 수 있으며 그 정보를 Transaction 으로 나타낸다.)을 서버의 공개키로 암호화하여 보내며 서버는 받은 코인의 이중사용 여부를 조사한 후 새로운 코인을 발급하여 2 단계때 받은 B 의 임시키로 암호화하여 보내준다. 3 단계에서 B 는 새로운 코인을 받은 후 4 단계에서 A 가 지급한 가격, 거래 ID, 날짜 등으로 구성되어 있는 영수증을 발급하여 1 단계에서 A 가 보낸 임시키로 암호화하여 보낸다.

2.2 PayWord

PayWord 는 인터넷에서 온라인상의 소액지불을 위한 프로토콜로써 일반적인 전자 상거래 프로토콜에서 자주 쓰이는 RSA 방식의 서명을 사용하는 대신 해쉬함수를 사용하여 계산 비용을 줄여 작은 거래에 적합하도록 설계하였으며, 전자 화폐가 온라인으로 서버와 통신한 것과는 달리 오프라인을 제공하여 브로커와의 통신을 줄인 것이다.[3]

사용자는 브로커에게 계좌를 열고 PayWord 의 인증서(Certificate)를 발급받은 후에 거래를 시작할 수 있다. 사용자는 보안이 보장된 채널을 통해 카드 번호, 공개키, IP 어드레스

의 정보를 보내면, 브로커는 브로커의 이름(B), 사용자이름(U), 사용자의 어드레스(A) 사용자의 공개키(P_K), 유효기간(E)과 그 밖의 정보(예를 들면 인증서의 일련번호, 크레디트 한계, 브로커에 대한 정보) 등이 들어 있는 인증서(Cu)를 브로커의 비밀키로 전자서명을 하여 발급해 준다. 이 인증서는 사용자의 공개키를 증명하는 것으로써 사용자가 전자서명을 한 Commitment 를 상인에게 보낼 때 상인은 이 인증서를 보고 Commitment 가 정당한 사용자에 의해 제작된 것임을 확인한다.

$$Cu = \{ B, U, A, P_K, E, Iu \}KR_B$$

인증서를 발급받은 사용자는 아래에서 보여지는 해쉬함수를 이용하여 역순으로 "paywords"라는 해쉬체인을 만든다. Wn 의 값을 먼저 선택한 후 Wn-1, Wn-1... W0 순으로 값을 발생시킨다. 이 해쉬 체인의 값은 거래를 할 때 이 값의 인덱스(Wm,M)와 함께 지불로써 1 번째 부터 N 번째 값까지 순서대로 이용된다.

$$Wi = h(Wi+1)$$

여기서 W0 의 값은 다른 해쉬체인 값처럼 지불로써 이용하는 것이 아니라 지불된 값들을 인증하는 값으로 사용된다. 사용자는 위의 체인의 값을 역순으로 만들었으므로 상인은 받은 값으로 다음 값을 예측할 수 없다. 그러나 받은 지불의 값은 단지 수와 그것의 인덱스로 이루어 진 것이기 때문에 상인은 받은 값이 유용한 것인지 확인해야 한다. 사용자는 이를 위해서 W0 값을 자신의 비밀키로 전자 서명을 해서 Commitment 를 만들어서 인증서와 함께 상인에게 보낸다. 상인이 (Wi, i)값을 받은 경우 상인은 우선 자신이 알고 있는 브로커의 공개키를 이용해서 사용자의 인증서(Cu)를 확인하고 그 중에서 사용자의 공개키를 꺼낸다. 그 공개키로 사용자가 보낸 Commitment 가 정당한 사용자가 만든 것인지 확인하고 Wi 값을 해쉬함수에 넣에 i 번 돌려서 Commitment 의 W0 값과 비교한다. 값이 일치하면 지불은 유용한 것이므로 서비스를 실시한다. 위와 같이 상인은 W0 값으로 지불이 유용한 지 검사하기 때문에 사용자가 만든 일련의 해쉬체인은 한 상인에게만 사용이 가능하다. 만약 사용자가 결재를 할 때까지 만든 체인의 값을 모두 사용하지 못했으면 나머지 해쉬체인의 값은 버리도록 한다.

그날의 거래를 모두 마친 후 상인은 사용자에게 받은 인증서와 마지막 지불(Wl, l)을 브로커에게 제출하며 브로커는 (Wl, l)를 검증한 후 사용자의 계좌에서 그만큼의 금액을 상인의 계좌에 입금한다.

3. 제안 프로토콜

1.1 프로토콜의 구성및 내용

거래에 참여하는 개체는 사용자(Customer), 상인(Vendor), 브로커(Bank)이며 프로토콜의 단순성을 위해서 토큰은 모두 같은 금액이라고 가정한다. KRa 는 A 의 비밀키, KUa 는 A 의 공개키를 뜻하며 H 는 MD5 해쉬함수이며 암호화된 내용은 {}로 표시하였다.

1) 상인과 거래

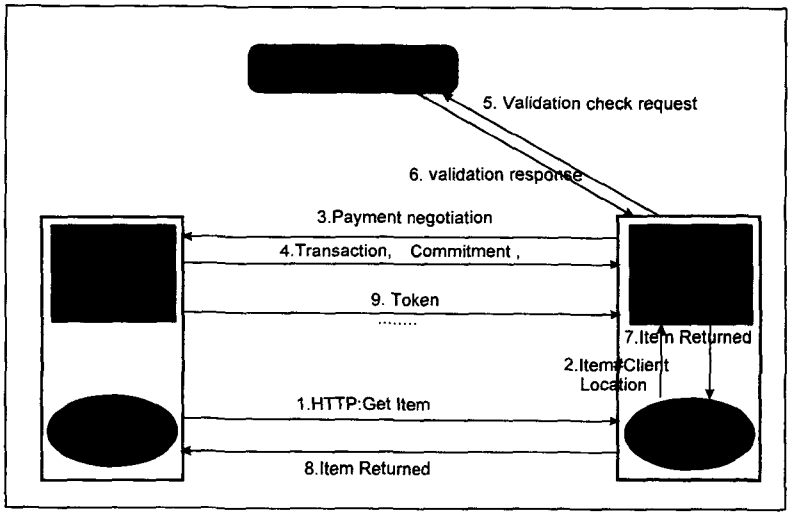
거래에 참여하는 개체는 각기 다른 신용도를 갖는다. 전자 지불 시스템은 브로커의 신용을 바탕으로 이루어 지기 때문에 브로커의 고의적인 사기는 없다고 가정하며, 상인은 회원제로 운영되기 때문에 중간의 신용도를 가지지만 언제든지 속임수의 가능성은 있다. 그러나 온라인 상의 사용자는 신용도가 없다고 생각되므로 상인이 서비스를 한 뒤 지불이 이루어지는 후불 형태는 불가능하다고 볼 수 있다. 그러므로 제안 프로토콜에서는 선불을 기본으로 거래가 이루어 진다고 가정하였다.

a. Negotiation

본 논문의 프로토콜은 온라인상의 유료 문서를 이용하는 목적으로 제안된 프로토콜이므로 사용자는 열람을 원하는 문서를 받아볼 때마다 상인이 책정한 금액을 지불한다고 가정한다. 그러나 사용자가 여러 문서를 계속적으로 열람할 때는 문서를 열람할 때마다 요금을 지불하는 것은 번거로운 일이 될 수 있으며, 속도면에서도 상당히 비효율적일 수 있다. 사용자는 열람할 문서의 값을 지불하기 전에 상인과 협상과정을 거쳐 지불 방식을 선택할 수 있다. 즉, 협상과정에서 사용자는 문서하나당 지불을 할 것인지, 몇개의 문서를 한번에 계산할 것인지를 선택한다. 사용자가 선불형식을 취한 경우 상인은 사용자가 지불한 금액을 기억하고 있다가 사용자에게 서비스를 할 때마다 금액을 감해 나가며 사용자가 그 금액을 초과해서 열람을 하려 할 때 메세지를 보내서 추가적인 지불을 할 것을 요구한다. 메세지를 받은 사용자는 연결을 끊던지 다시 협상과정을 거쳐 계속 문서를 열람하던지 한다.

b. 거래

사용자가 원하는 문서를 클릭하면 사용자의 IP 어드레스, 사용자가 클릭한 아이템의 정보가 상인의 서버에 전달된다. 상인은 사용자에게 협상의 메세지를 보내고, 사용자는 자신이 원하는 정보의 양에 따라 협상과정을 거쳐 지불형식을 선택하고 토큰과 Commitment을 상인에게 보낸다. 토큰은 전자화폐와 같이 실제적인 지불역할을 하는 것이며 Commitment는 지불에 사용한 토큰의 유용성을 증명하는 인증서이다. 사용자로부터 Commitment를 받은 상인은 브로커에 연결하여 Comimtment를 조사하여 받은 토큰이 유용한지 확인한다. 브로커는 Commitment를 조사해서 Commitment가 인증하는 토큰들이 이중사용이 아닌지 검사하고 상인에게 결과를 알려준다. 유용하다는 메세지가 오면 상인은 사용자와 거래를 시작한다. 인증서를 검증한 뒤에는 브로커와 온라인으로 통신할 필요가 없으며 상인은 토큰의 값만을 받으면 된다. 웹페이지당 토큰 하나의 금액이라고 가정하면 사용자는 웹페이지를 검색할때마다 토큰을 (W_1,1), (W_2,2) , ...의 순서대로 보내고 토큰을 받은 상인은 해쉬함수를 돌려서 유용성을 검증한다.



<그림 6> Web을 이용한 거래

2) 토큰발행과 Commitment

사용자가 거래를 시작하기 위해서는, 은행에서 현금을 찾는 것과 마찬가지로 브로커에게 토큰을 발행받아야 하며, 토큰을 발행받기 위해서는 하나의 난수(WN)를 선택해야 한

다. 이 난수는 해쉬함수를 이용하여 토큰을 만드는데 사용하는 시드(seed)값이며 이 값과 해쉬함수를 이용해서 역순으로 자신이 원하는 갯수 N 만큼의 해쉬 함수를 돌려서 Wn, Wn-1... W0의 해쉬체인을 만든다.

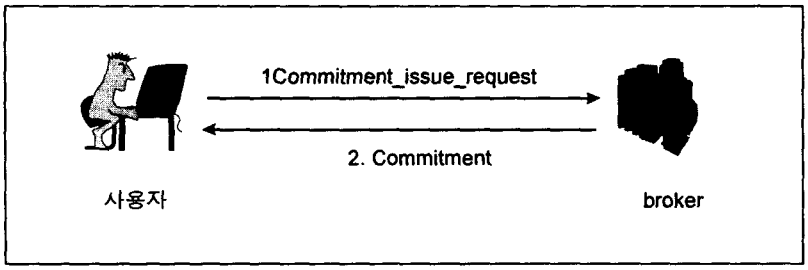
For i= n to 1

$W_{i-1} = h(W_i)$

토큰은 위의 함수를 이용하여 만든 Wn, Wn-1...값과 그것의 인덱스로 이루어 진다. 여기서 W0의 값은 토큰으로 사용되는 것이 아니라 토큰의 유용성을 검증하기 위한 루트의 값으로써 Commitment의 한 파라미터로 사용된다.

Token : (Wm,m)

Commitment는 토큰이 유용한 것임을 증명하는 인증서와 같은 역할을 한다. 사용자가 해쉬체인을 N에서 0값까지 모두 발생시켰으면 사용자는 체인의 루트값(W0)이 포함된 인증서 발행신청(Commitment_issue_request)을 브로커에게 보낸다. 인증서 발행신청에는 체인의 루트값뿐만 아니라 사용자의 ID, 금액, W0, N 등의 정보가 들어 있으며, 이 모든 값을 사용자의 비밀키로 서명하여 브로커에 보낸다.



<그림 5> 인증서 발행 신청

Commitment_issue_request = { CID, $, W0, Date, N, Nonce} KRu

CID	: 사용자의 ID
$	: 사용자가 요구하는 금액
W0	: 해쉬함수의 루트값
Date	: 발행신청 날짜
N	: 토큰 갯수

Nonce : 넌스값

브로커는 사용자가 인증서 발행신청에서 보낸 정보로 Commitment를 만들어 전송하고 그 금액만큼을 사용자 계좌에서 인출한다. 전자화폐 프로토콜에서 화폐발행서버가 직접 화폐를 제작해서 사용자에게 보내는 것과는 달리, 본 프로토콜에서는 브로커는 사용자가 직접 만든 일련의 토큰을 증명하는 Commitment만을 만들어 사용자에게 보낸다. 이 Commitment는 사용자가 만든 토큰이 합법적이며 나중에 상인이 토큰의 금액만큼을 상환받을 수 있다는 증명이므로 인증서의 발급은 중요하다. Commitment에는 해쉬체인의 루트값(**W0**), 체인의 수(**N**), 유효기간(**Expire_date**), 넌스, 서명한 브로커의 이름(**Broker_name**)등의 정보가 들어 있으며 브로커의 비밀키로 전자서명을 한다. 브로커는 사용자에게 발급한 Commitment 를 데이타 베이스에 저장한다.

Commitment = (W0,N, Expire_date, Nonce,Broker_name)

3) 토큰의 유용성 검증

본 논문에서 제안한 프로토콜은 위에서 언급한 바와 같이 거래를 시작할 때 사용자는 Commitment와 토큰을 상인에게 보내고 상인은 토큰의 유용성 검증을 위해 Commitment를 브로커에게 보낸다. 브로커는 Commitment를 받은 후 우선 자신의 전자서명인지 확인하고 Commitment안에 있는 루트의 값과 넌스값을 조사해서 이중사용이 아닌지 조사한다. 여기서 W0와 넌스값은 전자화폐에서의 일련번호(Serial number)와 같은 역할을 하기 때문에 데이타 베이스에서 Commitment의 W0값과 넌스값을 조사하여 값이 발견되지 않으면 이중사용을 한 것이므로 상인에게 거래를 취소하도록 한다. 제안 프로토콜에서는 Commitment만은 온라인으로 검증하고 Commitment에 따르는 토큰들은 오프라인으로 계산되므로 사용자가 동시에 두 상인에게 같은 Commitment를 사용할 수가 있다. 그러한 경우를 대비하기 위해 브로커는 인증의뢰가 온 Commitment에는 사용중이라는 플래그를 두어 표시한다. 상인이 브로커로부터 유효 메세지를 받으면 상인은 사용자가 보내는 토큰을 받고 원하는 문서를 제공한다. 토큰을 받을 때마다 상인은 해쉬함수를 돌려서 받은 토큰이 유용한지 확인해야 한다. 토큰값은 역수로 발생시킨 값이기 때문에 Commitment에 있는 해쉬체인의 루트값으로 토큰이 정당한지 검사할 수 있으며, 거래를 계속하여 그 날의 마감이 되면 상인과 사용자는 거래를 종료하고 상인은 사용자로부터 받은 토큰의 마지막 인덱스와 Commitment를 브로커에게 보낸다. 브로커는 토큰을 받은 후 해쉬함를 돌려서 받은 토큰이 유용한지를 조사하고 상인의 계좌에 토큰의 금액만큼을 예금한다. 브로커는 상인이

보낸 토큰의 Commitment를 데이타 베이스에서 찾아 Commitment의 값이 모두 사용된 것인지 사용이 덜 된것인지를 확인한다. 모두 사용한 것이면 데이타 베이스에서 그 값을 삭제하고 Commitment에 따른 토큰을 모두 사용하지 않았다면 사용한 토큰의 인덱스를 표시해 두어서 사용자가 다른 상인에게 다음 토큰부터 사용할 수 있도록 한다.

사용자가 상인에게 사용하는 토큰이 처음 인덱스가 아닌 경우 이미 사용한 토큰을 이중사용하는 것을 방지하기 위해 브로커는 상인이 Commitment의 인증을 의뢰할 때 사용가능한 인덱스를 상인에게 보내 준다. 상인은 사용자로부터 토큰을 받을 때 브로커로부터 받은 인덱스값과 비교해 보아서 유효한 범위내의 토큰인지 확인해야 한다.

3. 제안 프로토콜의 평가

제안 프로토콜은 다음과 같은 점으로 부하를 줄이고 소액거래에 적합하게 하였다.

- 사용자가 토큰을 만드므로 브로커의 부하를 줄일 수 있다.
- 상인이 토큰을 검증할 때 Commitment의 값만을 온라인으로 조사하므로 브로커와 상인의 통신 부하를 줄인다.
- 브로커가 각각의 토큰의 일련번호(Serial Number)를 모두 저장하지 않으므로 브로커의 저장데이타를 줄일 수 있다.
- 해쉬 함수를 이용하므로 RSA를 이용한 서명에 비교해서 계산 시간이 줄어든다.

제안 프로토콜은 기본적으로 PayWord의 알고리즘을 이용하고 전자화폐의 특성을 합쳐놓을 것이므로 두가지 프로토콜의 특성을 모두 가지고 있다. 다음은 두가지 프로토콜과의 비교를 통해 장단점을 분석해 보도록 한다.

1) 전자화폐와의 비교

제안 프로토콜에서는 브로커는 사용자가 토큰 발급을 위하여 Commitment 신청을 할 때 Commitment에 전자서명을 수행하고, 사용자와 상인이 거래를 시작할 때 Commitment만을 온라인으로 검증하므로 전자서명의 횟수와 통신의 부하가 현격히 줄어들게 된다. 특히 사용자가 한 상인과 계속적인 거래를 할 경우 상당히 효율적이다. 저장해야 할 데이타 양에 있어서도 토큰에 따른 인증서와 사용한 인덱스만을 기록하므로 저장해야 하는 양이 줄어든다.

주로 RSA 알고리즘을 이용하여 거래를 하는 화폐방식은 서로간에 공개키를 알아야

하며 키를 생성하거나 검증하는데 많은 시간이 걸리게 되지만[7][14] 제안 프로토콜은 사용자와 상인간에 서로의 전자서명을 사용하는 일이 없으며 각자는 브로커의 공개키만을 알면 된다.

그러나, 전자화폐는 어느 때나 다른 사람에게 양도할 수 있는 성격을 갖지만 제안 프로토콜에서는 한 인증서에 따른 일련의 토큰들을 거래가 시작되어 결재할 때까지 한 상인에게만 사용할 수 있으므로 제한된 양도성만을 지닌다. 또한 익명성의 제공은 전자화폐에서 상당히 중요한 문제가 될 수 있는데 제안 프로토콜의 경우 전자 화폐와 비교해 볼 때 아주 제한된 익명성만을 제공한다.

2) PayWord 와의 비교

PayWord 는 사용자의 신용을 기반으로 한 프로토콜이므로 사용자의 익명성이 전혀 보장되지 않는다. 사용자는 상인과 거래를 시작할 때 자신의 정보- IP 어드레스, 자신의 공개키-등을 Certificate 에 넣어 상인에게 넘겨주고 브로커와 결재를 할 시 브로커에게 사용자의 정보가 있는 Commitment 를 넘기므로 상인은 사용자가 누구인지 알 수 있으며 브로커는 사용자가 어떠한 상인과 거래를 했는지 알 수 있다. 제안 프로토콜에서는 사용자는 Commitment 에 자신의 정보를 포함할 필요가 없으므로 상인은 사용자가 누구인지 알 수 없으며 브로커가 Commitment 를 발급할 때 어떤 사용자에게 발급하였는지 기록하지 않으므로써 사용자의 익명성이 보장될 수 있다.

PayWord 에서 사용자는 Commitment 외에 자신을 인증하는 Certificate 를 가지고 있어야 하며 Commitment 에 자신의 비밀키로 전자서명을 해야 한다. 그러므로 상인은 Certificate 를 받아서 브로커의 공개키로 인증과정을 마친 뒤 Certificate 에 포함되어 있는 사용자의 공개키로 Commitment 를 인증해야 한다. 또한 신용이 좋지 않은 사용자가 계속적인 거래를 하는 것을 막기 위해서 Certificate 를 일정한 간격으로 재발급해야 하며 신용이 좋지 않은 사용자의 공개키를 모아 놓는 Hot-list 등을 관리해야 한다.

제안 프로토콜에서는 사용자가 미리 자신의 통장에서 토큰을 인출하는 형식을 사용하므로 신용이 없는 사용자들을 따로 관리할 필요가 없다 따라서 신용이 좋지 않은 사용자로 인하여 피해를 입는 상인이 생길 염려가 없으며, 상인이 사용자의 전자서명을 인증해야 할 필요가 없으며 브로커는 Certificate 를 발행하지 않아도 된다.

PayWord 는 한 상인에게 사용하는 payword 체인을 그날 모두 사용하지 않으면 남은 해쉬체인을 재사용할 수 없고, 다시 거래를 시작할 때 또 다른 payword 체인과 commitment 를 만들어야 한다. 그러나 제안 프로토콜에서는 한번 만든 체인은 그 자체에 가치를 포함하고 있으므로 사용하지 않은 인덱스부터 다시 사용할 수 있다. PayWord 에서 상인이 받은

체인을 결재할 시에는 사용자의 신용카드나 통장에서 사용한 금액만큼을 상인에게 넘겨줘야 하므로 모든 거래의 내역을 브로커가 어느 기간까지는 저장하고 있어야 한다. 사용자는 하루에 거래를 새로 시작하는 상인당 Commitment를 만들어야 하므로 브로커가 저장해야 할 Commitment의 양은 많아지게 된다. 그러므로 한번 만든 체인을 계속 사용하는 것은 브로커가 저장해야 할 인증서의 양을 줄이는 역할을 할 수 있다. 또한 제안 프로토콜은 PayWord 브로커에 계좌가 없는 사람일지라고 화폐를 양도받으므로써 전자 상거래를 할 수 있는 양도성을 부분적으로 제공한다.

Payword는 완전한 오프라인을 제공하므로 브로커와의 통신을 최소화하였다. 그러나 제안 프로토콜에서는 상인과 거래를 처음 시작할 때는 온라인으로 유용성 검증을 해야 하므로 PayWord에 비해서 커다란 부하로 작용할 수 있다. 최악의 경우 사용자가 한 상인에게 한번의 거래만 계속 할 경우에는 전자화폐와 같은 부하를 갖는다.

사용자가 만든 체인을 모두 사용할 수 있는 점은 장점으로 작용할 수 있지만 사용자는 남은 토큰과 사용한 인덱스를 자신의 시스템에 저장하여 계속 관리해야 하는 불편이 있을 수 있다.

4. 앞으로의 연구방향

제안 프로토콜은 한 상인에게 적은 금액의 계속적인 서비스를 받을 때 상당히 효율적이다. 사용자가 한 웹서버에게 집중적으로 계속적인 서비스를 받아야 할 때 전자화폐나 크레디트 카드 방식을 사용한다면 거래를 하는 개체 사이에 너무나 많은 정보를 주고 받아야 한다. 예를 들어 온라인상에서 증권투자 서비스 같이 하루에도 몇번씩 접근해서 정보를 받아보아야 하는 서버는 본 논문에서 제안하는 프로토콜이 효과적인 좋은 예이다.

제안된 프로토콜은 PayWord와 화폐의 성격을 함께 지녔기 때문에 각각의 장점을 가지고 있는 반면 두 프로토콜이 지녔던 장점의 많은 부분을 양보하였다. PayWord가 각각의 개체간에 통신을 최소한으로 한 반면 본 논문의 프로토콜은 전자화폐에 비해서는 통신이 현저하게 줄어들었으나 PayWord와 비교해서는 통신의 양이 많이 늘어났으며, 소액의 프로토콜을 위하여 제안된 것이기 때문에 계산의 양과 통신의 양을 줄이기 위해서 보안의 많은 면을 희생하였다. 그렇기 때문에 지불의 금액이 커지고 화폐를 가로채려는 사람이 많아진다면 위의 프로토콜은 상당히 취약해 질 수 있다. 전자 상거래의 보안의 정도를 높이기 위해서는 프로토콜이 복잡해 지는 것을 감수해야 하므로 사용하는 금액의 정도에 따라서 타협이 필요한 부분이라고 생각된다.

Reference

[1] NetBill, Carnegie Mellon University URL http://www.ini.cmu.edu/netbill/

[2] Steve Glassman, "The Millicent Protocol for Inexpensive Electronic Commerce"

[3] "PayWord and MicroMint- Two Simple Micropayment Schemes", RL Rivest, A Shamir, MIT

[4] Gennady Medvinsky and B. Clifford Neuman. NetCash: A Design for practical electronic currency on the Internet. In Proceedings of the First ACM Conference on Computer and Communications Security, November 1993.

[5] FV "First Virtual Over URL http://www.fv.com/

[6] "Micro Payment Transfer Protocol(MPTP) Version 0.1

[7] CyberCash Inc URL http://www.cybercash.com/

[8] M. Bellare, et al. iKP Family of Secure Electronic Payment Protocols URL http://www.zurich.ibm.com/Technology/security/extern/ecommerce

[9] Gennady Medvinsky and B. Clifford Neuman Requirements of Network Payment: The NetCheque Perspective. In Proceedings of IEEE Compcon'95, San Francisco, U.S.A., March 1995

[10] M. Peice, PayMe: Secure Payment for World Wide Web Services, Computer Science Department, Trinity College Dublin 2, Ireland May 1995

[11] D. Chaum "Achieving Electronic Privacy" Scientific American, August 1992, pp 96-101

[12] Ross Anderson "NetCard - A Practical Electronic Cash System", Computer Laboratory

[13]

[14] Cross-Industry Working Team, "Electronic Cash, Tokens and Payments in the National Information Infrastructure"

[15] Ecash URL http://www.digicash.com/

[16] Modex, URL http://www.mondex.com/mondex/

[17] 권도균 "WWW 보안과 전자화폐" URL http://wsp.nextel.net/seminar/etc