인터넷 상거래를 위한 효율적인 소액지불 시스템의 설계 및 구현

°성 원*, 조현성, 조현규, 함호상
충남대학교 컴퓨터공학과*, 한국전자통신연구원 전자거래연구팀

A Design and Implementation of Efficient Internet Micro-Payment System

° Won Seong*, Hyeon-Sung Cho, Hyun-Kyu Cho, Ho-Sang Ham
Computer Engineering Department, Chungnam National Univ*
Electronic Commerce Dept.,Electronics and Telecommunications Research Institute.

요약

최근 인터넷을 매개체로 하여 디지털 정보 상품들의 거래를 행하는 새로운 지불 시스템이 요구되고 있다. 이는 기존의 지불 시스템들을 디지털 정보 상품 거래에 적용시키기 어렵기 때문이다. 기존의 지불 시스템들은 어느 정도의 큰 액수를 다루는 대액지불 방식들이므로 각각의 개별 정보들의 거래를 위해서 사용되어진다면 정보 상품 거래의 이익보다 운영 비용이 더 크므로 경제성이 없게 된다. 그러하여, 최근에 인터넷에서의 소액의 디지털 정보 상품들의 거래에만 사용되어질 새로운 소액지불 시스템들이 연구되고 있고 몇 가지 방식들이 제안되었다. 그러나, 제안된 소액지불 시스템들은 불필요한 계좌 생성,관리와 지불된 모든 소액 화폐의 집합(aggregation),저장이라는 문제점을 가지고 있다. 이들 문제점들은 많은 부담을 주는 전자사인 등과 함께 소액지불 시스템의 운영비용을 크게 늘리는 항목이다. 그러므로, 효율적인 소액지불 시스템은 적적한 보안, 값싼 핵심 메커니즘의 구성과 함께 이들 문제점들을 해결해야만 한다.

이에 본 연구는 인터넷 전자상거래를 가능하게 해 줄 소액대금결제 방식인 PayHash 시스템을 설계하였다. 이 시스템은 값싼 화폐 생성 메커니즘과 적절한 보안을 가지고 있고 필요 없는 계좌 생성과 지불 값들의 집합, 저장을 피함으로써 소액지불 시스템으로서의 조건을 만족시키고 있다.

1. 서론

인터넷은 디지털 개별 정보의 유통을 가능하게 해주는 매개체이다. 이때 거래 유통되는 디지털 정보의 종류로는 "날씨 정보", "신문기사","그림 파일 한장", "영화의 한 장면","음악화일 한 개", 등등이 있을 수 있다. 그러나, 위와 같은 개별 정보의 유통은 종전과 같은 대금결제 방식으로는 불가능하다. 이유는 위와 같은 개별정보의 가격은 상당히 적은 몇 십원 정도의 소액이 될 것인데 기존의 지불 방식으로는 운용 비용이 더 클 것이므로 결제성이 없기 때문이다. 그러므로, 인터넷을 통해서 개별 정보들을 유통시킬 수 있는 새로운 대금 결제 메커니즘인 소액지불 방식이 필요하게 된다.

최근에 연구되고 있는 기존의 소액대금결제 프로토콜로는 "Millicent"[2],"PayWord"[1],"MicroMint"[1],"iKP"[3][4] 등이 있다. 이 들 각각의 프로토콜들은 나름대로의 방식으로 안전성과 소비용화를 추구하고 있다. 그러나, 이들 방식들은 핵심 메커니즘의 소비용화와 적절한 보안의 만족 등에는 문제가 없었으나 "계좌(account)의 관리"와 "지불 값들의 집중"에는 문제점을 가지고 있다. 사실 소액지불 시스템에서는 계좌 관리와 지불 값들의 집중의 합리적인 해결이 가장 먼저 고려되고 처리되어져야 할 항목이다. 이러한 문제가 운영비용을 늘리고 시스템을 복잡하게 만들기 때문이다.

소액지불을 통해서 구입하는 상품은 대부분 정보 상품들이다. 이러한 소액 상품들은 사탕 자판기에서 구매할 수 있는 사탕으로 비유될 수 있다. 이때, 지불은 고객이 소액의 동전을 가지고 행하게 될 것이다. 그런데, 이때, 자판기나 자판기의 관리자가 각각의 지불에 대해 상세하게 "구매자는 누구고 구매 액수는 얼마다"라고 꼼꼼히 적어 기록하지는 않을 것이다. 그러다가, 또한 몇 개의 사탕을 팔고 구매하는 과정이 큰 액수의 지폐를 다루는 과정보다 더 복잡하고 비용이 더 든다면 모순이다. 이 비유와 마찬가지로 소액지불 시스템은 사탕과도 같은 소액의 정보 상품들의 거래에 사용되어진다. 그렇다면, 이 같은 소액지불 시스템들은 운영 비용의 최소화라는 원칙 아래 모든 시스템이 운영되어져야만 한다.

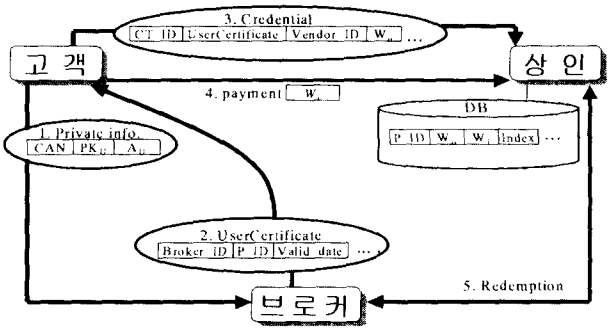
그리하여, 본 논문에서 설계, 구현한 소액지불 시스템 PayHash 는 운영 비용의 저렴화, 적절한 보안의 만족, 지불 값들의 집중 문제의 해결, 불필요한 계좌의 사용 배제 등을 만족시킴으로써 효율적인 소액지불 시스템으로서 설계되었다.

2. PayHash

본 장에서는 PayHash 소액대금결제 프로토콜의 동작 원리에 대해서 간략히 알아본다.

PayHash 는 수행 중 가능한 한 모든 부분마다 해쉬 기능(hash function)을 적용시켜 지불마다 요구되는 공용키(public key) 사용의 수를 최소화 시키고 또한 브로커와의 접촉을 최소하 시켜서 비용을 저렴화 시킴과 동시에 일방향 해쉬 함수[6]의 능을 이용하여 값싸고 빠른 지불방식을 채택하고 있다. 또 다른 소액지불 시스템들이 모든 지불 값들을 전부 저장 보관하는 점을 개선함으로써 지불 값들의 집중 관리의 문제점 해결하고 있다. 또, 필요한 만큼의 계좌들만을 생성 관리하 있다.

PayHash 는 크레디트-기반의 시스템이며 "payhash"라는 해쉬 값들의 사슬메커니즘에 기반을 두고 있다. 소액지불을 위하여 해쉬 기능을 이용하는 이러한 아이디어의 응용은 Anderson[7] 과 Pederson[7]에 의해서도 독립적으로 행해져 왔다. 이 모델의 참여 party 들은 브로커들(Brokers), 고객들(Users), 그리고 상인들(Vendors)이다. 전체적인 시스템의 흐름은 [그림 1]과 같고 각 party 별 동작 관계는 다음과 같다.



[그림 1] 시스템의 전체 흐름도

고객 U 가 브로커 B 에게 자신의 계좌(,account)와 거래시 자신의 존재를 증명시켜 줄 "고객 인증서(UserCertificate)"를 요구함으로써 두 party 간의 관계가 시작된다. 이는 고객의 상거래 참여의 시작을 의미한다. U 는 안전한 채널을 통해서 B 에게 자신의 크레디트-카드 번호(CAN), 공개키 PK_U, 그리고 자신의 배달 주소 A_U 등을 준다. 이 후 고객이 써버린 payhash 지불들은 그의 크레디트 카드 구좌로 부과된다. 그의 배달 주소는 그의 인터넷/전자우편 주소다; 고객의 인증서는 고객이 보내온 위에서와 같은 고객의 정보들을 가지고 B 가 만들어서 다시 고객에게 주는 것으로서 이 고객이 정당한 존재임을 증명한다는 정보들을 담고 있다. 아래에서 고객을 나타내는 P_ID 는 B 가 자신이 임의로 생성해낸 임의값 R_C 와 고객의 크레디트 번호를 합쳐서 해쉬 함수에 적용시킨 결과값으로서 고객의 의사 아이디(Pseudo_ID)가 된다. 이 P_ID 는 다른 소액지불 시스템들이 갖추지 못한 익명성의 특성을 갖게 해주는 데 쓰인다.

$$P\_ID = H(R_C, CAN)$$
$$C_U = \{B, P\_ID, A_U, PK_U, E, I_U\} SK_B$$

위 식에서 $\{\}SK_B$ 는 {}안의 내용을 B 의 비밀키로 전자 서명한다는 것을 나타내며, 인증서(UserCertificate)는 만기일 E 와 인증서의 시리얼 번호, 각 상인들에게 주어질 수 있는 신용한계 등의 정보 등도 I_U 에 담고 있다. 그래서, 고객의 인증서는 위 식의 C_U 의 형태를 가진다. 이 C_U 는 고객-상인 간의 지불이 행해질 때, Credential 의 내부에 포함되어져서 쓰이게 되는데 U 가 구매를 하고 지불을 하기에 앞서서 Credential 를 V 에게 제시하게 된다. 이때, V 가 자신이 알고 있는 B 의 PK_B 를 가지고 C_U 를 복호화 시켜 그 Credential 의 내부를 열어서 보면 해당 U 가 만기일까지 B 에 의해서 정당한 존재로서 인정되고 있는지의 여부를 확인할 수 있다.

이제 실제로 고객이 상인의 웹사이트를 방문하여 몇 가지 정보 상품을 구입하는 단계에 이르게 된다. U 가 새로운 상인 v 에 접촉했을 때, U 는 루트(root) W_0 를 가지고 연속적으로 해쉬 함수를 적용시켜 새로운 payhash 사슬 $w_1,\ldots,w_n$ 을 생성하게 된다. 여기서 n 은 고객이 편리를 위해서 선택되어지고 고객은 그 사슬을 위한 Credential(아래에서 CT)를 다음과 같이 계산한다.

$$CT = \{V, C_U, w_0, D, I_M\} SK_U$$
$$CT\_Sig = H\{P\_ID + CT\}$$

여기서 V 는 상인을 나타내고 C_U 는 U 의 인증서, w_0 는 payhash 사슬의 루트, D 는 유효 날짜, I_M 은 부가적인 정보들(예를 들면, 사슬의 길이 정보 등)을 나타낸다. CT 는 U 에 의해서 비밀키로 사인되어 V 에게 보내진다. V 는 M 에 씌워진 U 의 사인(signature)을 검사하고 C_U (M 에 포함되어져 온)에 씌어진 브로커의 사인도 검사한다. 이를 통해 V 는 U 의 존재를 믿게 되고 이후 행해지는 U 의 지불을 받아들이게 되는 것이다. V 는 U 의 부정을 조사하기 위해서 그날의 끝까지 Credential 를 저장해야만 한다. 그러나, 해쉬 사슬 메커니즘에 기반한 다른 유사 시스템들은 날마다 본 시스템의 Credential 같은 특정 상인에만 유효한 지불 증명서를 만들어내고 그를 기준으로 중복되게 계좌를 만들어 관리를 한다. 그러나, 이는 소액지불 시스템으로서는 큰 부담이 아닐 수 없다. 그러하여, 본 시스템에서는 V 가 고객 U 의 Credential 의 내부를 열어서 알아낸 P_ID 를 기준으로 계좌를 만들고 여러가지 Credential 마다의 계좌는 생성하지 않는다. 또한 CT 를 형성하는 SK_U 로의 전자서명도 처음 Credential 의 생성시에만 한번 하게 되고 후일부터는 CT_Sig 를 사용, 전날의 Credential 를 확인하는 방법으로 전자서명의 부담도 줄이고 있다.

U 와 V 는 지불되어져야 할 양에 대해서 동의할 필요가 있는데 전형적인 소액지불 프로토콜에서는 상당히 적은 양을 다루게 되고 기본적인 양들을 합한 덩어리들도 취급할 수 있다. U 로부터 V 로 행해지는 지불 P 는 하나의 지불 값과 그것의 인덱스로 구성되어 진다.

$$P = (w_i)$$

첫번째 V 로의 지불은 U 와 관계되는 Credential 의 동봉과 함께 이루어지면 이후의 지불은 단지 payhash 값과 그 인덱스만을 가지고 행한다. 이 때, 여타 소액지불 시스템 제안들은 V 가 U 에게 정보 서비스를 제공하고 받은 모든 지불 화폐들을 저장하고 있게 된다. 그러나, 이는 경제적인 측면에서 바람직하지 않다. 본 시스템은 해쉬 함수의 일방향성을 이용 V 가 U 의 P_ID 계좌에 w_0 와 가장 최근에 지불된 payhash 값만을 저장시킴으로써 V 가 받은 모든 지불 값들을 저장시키는 문제점을 해결하고 있다. 한 상인 V 는 B 와 미리 먼저 관계를 가지고 있을 필요는 없다. 그러나, 믿을 수 있는 채널을 통해서 B 의 공개키 PK_B 를 얻어 놓을 필요가 있다. 그래야만 V 는 B 에 의해서 사인된 인증서를 검사할 수 있다. V 는 U 로부터 모은 payhash 들을 B 에게 되팔아(redemption) 올 수 있어야 한다. 일반적으로 그 날의 끝에 V 는 B 에게 redemption 메시지를 보내게 되는데 이 메시지는 U 로부터 받은 C_U 와 마지막 지불 P 로 이루어진다.

3. PayHash 구성 메커니즘

3.1. hash 일방향성을 이용한 Coin 생성 메커니즘

해쉬(hash) 함수는 일방향의 성질을 가지고 있는데 어떤 주어

진 값을 해쉬 함수에 적용시켜 결과 값을 얻어내기는 쉬우나 반대로, 원래의 주어졌던 값을 역으로 계산해 내기는 어려운 특징을 가지고 있다. 이것은 값싸면서도 유용한 보안 기술이다.[6]

본 연구에서 구현한 소액지불 프로토콜의 방식은 PayWord[1], NetCard[5]에서의 제안 방식에 근간을 두고 있다. 각각의 coin을 사용하기 위해 고객이 해쉬 함수를 이용해 coin 들의 stick을 만들고 그 stick 에 사인해서 상인해서 보낸다. stick 에 들어갈 coin 의 생성은 다음과 같은 과정에 따른다.

$$w_i = h(w_{i+1}) \quad (i = n-1, n-2, n-3, \ldots, 0)$$

이제, 고객은 부가적인 정보들과 함께 w0 를 상인에게 보내고 그 후에 고객은 stick 에서 차례로 각 coin 들을 꺼내서 지불하고 싶은 만큼 상인에게 보내고, 상인 간단한 해쉬 함수로 계산 해봄으로써 각 coin 의 유효성을 검증할 수 있는 방식이다. 이 방식은 hash 적용의 소비용화로 인하여 소액지불 프로토콜의 구현에 가장 적합한 것으로 이해된다.

3.2. 계좌(account),지불 집합(aggregation)의 해결

최소한의 계좌 수 유지와 지불 값들의 집합, 저장 문제의 해결은 소액지불 시스템들이라면 꼭 해결해야 할 문제다. 대액지불 시스템의 경우라면 분명 모든 지불 값들을 저장, 기록하여야만 할 것이고 시스템의 신뢰성 있는 운영에 필요하다면 여러 가지 계좌의 생성, 운영도 합리적이라고 할 수 있다. 그러나, 소액지불 시스템의 경우엔 고객의 하나의 지불아라는 것이 너무나도 적은 소액이므로 수십, 수백번의 지불을 합쳐도 대액지불 한번의 액수보다도 못할 수 있다. 그런데 이렇게 행해지는 모든 소액의 지불 값들을 각 참여자들이 지불 확인을 위해서 모두 집합,저장,유지시켜야 하고 보안을 위해서 여러가지 계좌를 생성, 유지하며 여러 번의 전자 사인을 행한다면 비경제적이다.

이에 본 PayHash 는 모든 고객이 지불한 모든 비불 값들을 상인이 모두 저장하지는 않고 상인이 고객의 유사 아이디를 기준으로 한 계좌만을 유지시켜 거기에 w_0 와 현재 받은 w_i 값, 그 값의 인덱스를 기록하는 방식을 사용한다. 이렇게 하여 상인이지불 값들을 모두 저장하는 부담을 줄이고 있다.

$$CT\_Sig = H(P\_ID + Credential)$$

PayHash 는 전체 시스템의 처리 부담을 크게 하는 전자사인의 문제도 해결하고있다. 처음에 생성시키는 Credential 의 경우에만 고객의 저자 사인을 하여 상인이 확인할 수 있도록 하고 그 후일 부터는 Credential 의 값과 P_ID 를 합쳐서 해쉬한 값(위 식의 CT_Sig)만을 전자사인으로서 제시함으로써 전자 사인의 시스템 부담을 줄이도록 했고 한 고객이 익사 아이디에 기준한 한 계좌만을 갖을 수 있도록 함으로써 불필요한 지불 증명서에 기준한 계좌 생성 관리도 없앴다.

4. 소액지불 시스템 비교

본 장에서는 최근에 제안된 소액지불 시스템들과 본 논문에서 제안한 PayHash 방식을 몇 가지 항목에 대하여 비교한다.[8] 비교 항목은 소액지불 시스템에서 중요시해야 할 부분들로서 계좌(account)관리, 지불집합(aggregation)처리, 보안강도, 익명성(anonymity) 등이다.

	PayHash	PayWord	Millicent	iKP
보안 강도	▲	▲	▲	●●
메커니즘 간결성	●●	●●	▲	××
Aggregation 처리	●	▲	×	×
계좌 축소	●	▲	×	×
익명성	▲	×	×	●●

●: 좋음, ▲:보통, ×:나쁨

[표 1] 소액지불 시스템 비교

[표 1]에서 보듯이 PayHash 시스템은 해쉬 사슬 방식에 기반을 둠으로써 PayWord 와 같은 수준의 보안강도와 메커니즘 간결성을 갖추고 있고 aggregation 처리와 계좌 관리의 부담을 크게 줄이고 있다. 또한, 의사 아이디(Pseudo ID)의 사용을 통해 부분적인 익명성도 갖추고 있다.

5. 결론

소액대금결제 시스템이 제대로 만들어지기 위해서는 시스템 전반적인 보안과 핵심 메커니즘의 안전성의 추구도 중요하겠지만 무엇보다도 합리적인 소액대금지불을 성립시킬 수 있는 시스템 유지비용의 저렴화가 이뤄져야 한다. 이는 소액대금결제 시스템들은 아주 적은 소량의 금액을 다루는 특별한 성질을 가지고 있으므로 일반적인 대액지불 시스템들에서와 같은 값비싼 보안장치의 마련만이 최선은 아니기 때문이다. 보안 조건들을 모두 충족시킨다는 것은 그 만큼 유지비용을 늘어나게 하기 때문에 가장 적절한 보안 조건들만을 만족시키면서 프로토콜이 이뤄져야 한다. 그러므로, 만족시키고자 하는 보안 조건의 구현도 최소의 비용만을 가지고 실현시킬 수 있는 적절한 메커니즘을 가져야 함은 물론이거니와 특히 불필요한 계좌의 생성과 관리를 줄이고 지불 값들의 집합 관리를 되도록 피해야만 한다.

이에 본 연구에서는 운영의 효율성을 증대시킬 수 있는 방안들을 기반으로 인터넷 상거래를 위한 개선된 소액지불 시스템을 JAVA 를 통해 구현하였다.

참고 문헌

[1] R.L.Rivest and Adi Shamir,"PayWord and MicroMint :Two simple micropayment schemes", Available from authors,May 1996.

[2] Steve Glassman and Mark Manasse,"The Millicent Protocol for Inexpensive Electronic Commerce",http://www.millicent.digital.com

[3] R.hauser, M.Steiner, and M.Waidner,"Micro-Payments based on iKP",Available from authors,Dec.1995

[4] M.Bellare,J.Garay, and M.Waidner, "iKP – A Family of Secure Electronic Payment Protocols", Available from authors,July 1995

[5] R.Anderson, H.Manifavas, and C.Sutherland,"A Practical electronic cash system", Available from authors, 1995

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

[7] Torben P.Pederson,"Electronic payments of small amounts" Technical Report DAIMI PB-495, Arthus Univ.Aug 1995

[8] 성원,공은배, "안전한 소액지불프로토콜의 비교.검토와 구현",한국통신학회 NCS'97, pp123-126, Dec. 1997