개선된 Payword 프로토콜

강도근, 최종훈, 공은배 충남대학교 컴퓨터공학과,
조현성, 조현규 한국 전자통신연구원
{dkkang, jhoon, keb}@comeng.chungnam.ac.kr, {hsc}@etri.re.kr

Revised Payword Protocol

DoKeun Kang, Jonghoon Choi, EunBae Kong Computer Engineering Department, Chungnam National Univ.
Hyeonsung Cho, Hyunkyu Cho Electronics and Telecommunications Research Institute

요약

Payword 프로토콜은 전체적인 프로토콜의 수행 과정이 복잡하지 않고, 과도한 암호화 모듈을 사용하는 대신 단방향 해쉬 함수를 사용함으로써 전체 시스템의 속도가 빠르며, 대금을 지불하고 정산하는 방식에 있어 상당한 효율성을 가지고 있어 소액대금결제시스템으로 안성맞춤이다. 그러나 Payword 프로토콜은 프로토콜의 구조적인 결함으로 생길 수 있는 문제점을 가지고 있다. Payword 프로토콜에서 사용자는 상거래 행위에 참가하기 위해 Broker 에게 계좌요구정보를 전송한 후, 그 결과로 Certificate 를 받는다. 사용자는 전송 받은 Certificate 를 기반으로 Commitment 를 생성하고 이를 Vendor 에 전송한다. Vendor 는 Commitment 에 기반으로 하여 일정 기간동안의 사용자의 Payword 체인은 정당한 것으로 간주함으로 상거래 행위가 생긴다. 따라서 매우 빠르고 효율적인 상거래 행위가 진행될 수 있다. 그러나 사용자는 여러 Vendor 와 거래를 하기 위해서는 Commitment 를 각각의 Vendor 에 전송하여야 한다. 이점을 악용하여 사용자는 자신이 Broker 와 사용하기로 약정한 Payword 체인을 여러 Vendor 에서 전부 사용할 수 있는 문제가 발생하게 된다.

본 논문에서는 이러한 사용자에 의한 악의적인 Payword 의 사용을 방지하고, 보다 소액대금결제 시스템에 적합한 Payword 프로토콜을 개정한 Payword 프로토콜을 제안한다.

1. 서론

지금까지 행해지던 정보 유통의 특징은 한마디로 "유통 정보의 번들(bundle)화"로 설명될 수 있다. 그러나, 앞으로 인터넷을 이용하는 정보 유통은 "개별 정보 유통의 확산"으로 설명할 수 있다. 예를 들면, 지금까지는 소비자가 잡지의 특정 부분을 읽고자 하더라도 잡지 한 권을 구입해야 했지만, 인터넷 기술의 발전과 대중화로 이러한 잡지의 특정 기사를 구입할 수 있게 되었다.

이러한 개별 정보의 유통은 종전과 같은 대금결제 방식으로는 불가능한데, 그 이유는 이러한 개별정보의 가격은 10 원 이하의 소액이 될 것인데 기존의 지불 방식으로는 운용 비용이 더 클 것이므로 경제성이 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. 이러한 각각의 프로토콜들은 나름대로의 방식으로 안전성과 소 비용화를 추구하고 있다. 그 중 Payword 프로토콜은 프로토콜의 핵심 메커니즘이 매우 효율적이고 속도도 빨라 소액대금결제에 알맞은 것으로 볼 수 있다. Payword 프로토콜의 주요 목적은 수행 중 가능한 한 모든 부분마다 해쉬 기능(hash function)을 적용시켜 지불마다 요구되는 공용키(public key) 작용의 수를 최소화 시키고 또한 브로커와의 접촉을 최소화 시켜서 소액지불을 가능하게 하자는 데 있다.

Payword 프로토콜에서는 지불 정보의 표시를 "PChain"이라는 것으로 하게 되는데, 이 PChain 체인은 단방향 해쉬 함수로 계산하여 "연속적인 체인형태(chain)"로 만들어 이용한다. 사용자는 상인에게 필요한 정보들과 첫번째 PChain 값을 주게 되고 필요한 서비스를 받기 위해서 연속적인 PChain 값을 보내게 되고 상인은 미리 받은 PChain 값과 나중에 보내온 PChain 값을 가지고 해쉬 함수 적용으로 맞추어 본 후 상품 전달 서비스를 해 주는 방식이다. 이 프로토콜은 바로 단방향 해쉬 함수[6]의 기능을 이용하여 값싸고 빠른 지불 방식을 채택하고 있는 것이다.

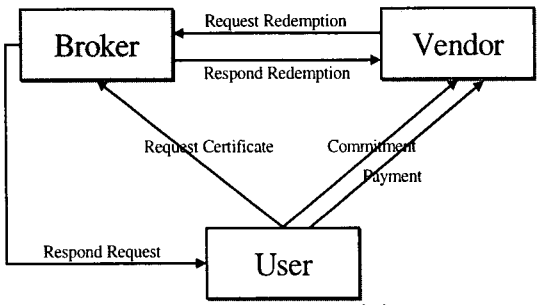
본 논문에서는 Payword 프로토콜의 취약점을 제시하고 이를 개선한 Payword 프로토콜을 제안한다. 2 장에서는 기존 Payword 프로토콜에 대한 개략적인 설명과 발생할 수 있는 취약점을 설명하고, 3 장에선 이 취약점을 개선한 개선된 Payword 프로토콜에 대한 설명을 하며 4 장에서는 결론과 향후과제에 대해 논하기로 한다.

2. Payword 프로토콜 및 취약점

Payword 프로토콜은 신용-기반(지불에 참가하는 파티(예 은행)가 믿을만한 정보를 제공한다는 것을 가정하고 동작하는)의 시스템이며, 지불 정보의 표현을 "Payword"라는 Hash 값들을 일정한 체인형태(이하 PChain)로 표현한다. Payword 프로토콜에서 지불 정보을 표현하는데 사용되는 단방향 해쉬 함수의 특성은 f(x)의 결과를 얻어내기는 쉬우나, f(x)의 결과를 만드는데 필요한 x 값을 찾아내는 것은 무척 어렵다는 것에 있다. Payword 프로토콜에서는 User, Vendor, Broker 의 3 가지 파티가 있으며, User 는 사용자를 의미하고, Vendor 는 아이템을 제공하는 업체나 Web Server 를, Broker 는 사용자가 자신의 Account 나

신용카드의 정보를 관리하는 Agent 나 은행을 의미한다. 각 파티들의 구성과 동작은 [그림 1]과 같이 구성된다.

Payword 프로토콜의 전체적인 흐름을 살펴보면 다음과 같다. 사용자는 상거래에 참가하기 위해서는 Broker 로부터 계정정보에 대한 보증서인 Certificate 를 발급 받아야 한다. 사용자는 안전한 경로를 통하여 계정정보에 필요한 정보를 사용자의 공개키로 암호화 하여서 Broker 에게 준다.



[그림 1] Payword 프로토콜 전체 흐름도

계정등록정보를 받은 Broker 는 사용자가 보내온 정보가 정당하면 계정정보에 대한 보증서인 Certificate 를 사용자에게 발급한다. 이 Certificate 는 사용자가 Vendor 의 아이템을 구입하고 난 후, 대금을 지급하기 위해서 Vendor 에 Commitment 를 제시하고, PChain 을 만들어 Vendor 에게 전송하게 되는데 이 Commitment 를 만드는데 반드시 필요한 정보이다. 이 'Commitment'는 사용자가 보내는 Commitment 를 받는 Vendor 에게, Commitment 가 Broker 로부터 정당하게 발급을 받은 Certificate 를 기반으로 생성되었으며, Commitment 를 기반으로 사용자가 생성하는 PChain 이 정당하다는 것을 증명하는 정보이다. Commitment 를 전송 받은 Vendor 는 Commitment 의 정당성을 증명해 본 후 정당하다고 판단이 되면, 그 이후로 특정 기간(혹은 시간) 동안 Commitment 를 제출한 사용자의 PChain 는 전부 정당하다고 가정을 하고 사용자가 구입한 아이템을 사용자에게 배달하게 된다.

Vendor 는 사용자로부터 받은 PChain 을 데이터베이스에 모아두었다가, 적절한 시간이 되면 모아두었던 PChain 에 해당하는 만큼의 결제대금을 Broker 에게 요청할 수 있다. Vendor 는 Broker 에게 사용자로부터 받은 Commitment 에서 추출한 Certificate 와 사용자로부터 받은 PChain 을 같이 제시한다. Broker 는 Vendor 가 보내온 Certificate 를 가지고 Vendor 가 대금을 요청하는 사용자의 정당성을 검사한다. 정당성이 입증이 되면, Vendor 가 보낸 PChain 을 검사해서 정당하면 Vendor 가 보내온 PChain 의 양 만큼을 사용자의 계정으로부터 Vendor 에게 지급한다

Payword 프로토콜은 전체적인 프로토콜의 수행 과정이 복잡하지 않고, 과도한 암호화 모듈을 사용하는 대신 단방향 해쉬함수를 사용함으로써 전체 시스템의 속도가 빠르며, 대금을 지불하고 정산하는 방식에 있어 상당한 효율성을 가지고 있어 소액대금결제시스템으로 안성맞춤이다. 그러나 Payword 프로토콜은 프로토콜의 구조적인 결함으로 생길 수 있는 문제점을 가지고 있다. Payword 프로토콜에서 Vendor 는 사용자가 전송한 Commitment 을 기반으로 하여 일정 기간동안의 사용자의 Payword 체인은 정당한 것으로 간주함으로 상거래 행위가 생긴다. 그러나 사용자는 여러 Vendor 와 거래를 하기 위해서는 Commitment 를 각각의 Vendor 에 전송하여야 하여야 하는 점을 악용하여 사용자는 자신이 Broker 와 사용하기로 약정한 Payword 체인을 여러 Vendor 에서 전부 사용할 수 있는 문제가 발생하게 된다. 따라서 Vendor 는 Broker 에게 대금 결제를 요구할 때, 사용자에게 전송받은 PChain 만큼의 손해를 볼 수 밖에 없다.

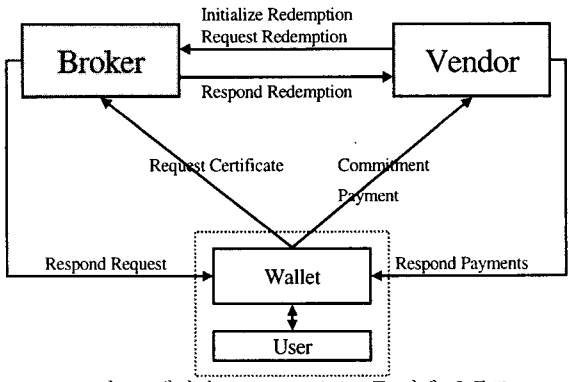
3. Payword 프로토콜의 개선

소액대금결제 시스템은 인터넷 상에서 아이템에 대한 대금처리를 위해 고안되었다. 따라서 사용자가 Web 브라우저를 통하여 아이템을 구입하는 경우, 아이템 구입에 필요한 데이터를 사용자가 일일이 입력하는 것은 너무 불편한 방식이며, CGI 를 통하여 사용하는 것은 사용자의 개인적인 정보와 데이터가 인터넷으로 공개가 되므로 보안상 문제가 된다. 따라서 이를 사용자의 정보를 관리하고 처리해 줄 수 기능을 담당하는 개체가 필요하다. 또한 Payword 프로토콜에서는 사용자가 동일한 Commitment 를 동시에 여러 Vendor 에 제출하고, Broker 와 약정한 한도액을 전부 소진하는 악의적인 사용자의 행동을 막을 방법이 없다.

위 두 가지를 개선하기 위해 원래의 Payword 프로토콜에 사용자의 정보를 관리하고 사용자의 악의적인 PChain 의 사용을 막기 위하여 Wallet 이라는 개체를 추가하였다. 이 Wallet 은 다음 두 가지의 기능을 담당한다.

- 사용자 정보의 관리 : 사용자가 Payword 프로토콜을 진행할 때 입력하고 처리해야 할 정보를 Web 브라우저를 통하여 관리한다.
- 사용자의 PChain 사용 관리 : 사용자의 악의적인 PChain 의 사용을 막고, 정당한 PChain 사용을 보장하는 기능을 한다.

Wallet 이 추가된 Payword 프로토콜의 구성과 전체 흐름도는 다음과 같다. Payword 프로토콜에서의 사용자의 부분이 크게 사용자의 정보를 관리하고 기능을 처리하는 Wallet 과 Wallet 을 사용하는 사용자로 구분된다. 사용자는 단지 Wallet 에 필요한 정보를 제공하고, Wallet 이 실질적인 모든 프로토콜에 필요한 정보를 관리하고 기능을 수행하는 역할을 담당한다. 따라서 사용자는 복잡한 정보의 처리 및 관리에 신경을 쓰지 않게 되고, 단지 프로토콜이 진행될 수 있는 정보만 제공하면 된다.



[그림 2] 개선된 Payword 프로토콜 전체 흐름도

3.1 PChain 사용방지

Wallet 은 사용자의 정보 관리 뿐만 아니라 사용자가 악의적

인 PChain 의 사용을 방지한다. 사용자는 Broker 에 Request Certificate 를 요구하고 Broker 로부터 Certificate 를 전송 받는다. 사용자가 아이템을 구입하려 할 때, 사용자는 Wallet 에게 Commitment 정보를 생성하여 이를 Vendor 에 전송토록 요구한다. 기존의 Payword 프로토콜과는 달리, Wallet 은 사용자가 Broker 와 약정한 금액의 전체에 해당하는 액수 만큼의 Commitment 를 생성치 않고, 사용자가 사용한 현재의 PChain 의 잔여량 만큼을 가지고 Commitment 를 생성하여 Vendor 에 전송하게 된다. 따라서 악의적인 사용자가 Commitment 를 여러 Vendor 에 전송하여 PChain 을 전부 소비하는 문제를 방지한다. 그러나 이 경우 Wallet 은 항상 사용자가 사용하는 PChain 에 대한 정보를 유지하여야 하고, 이 정보를 사용자가 메모리 접근과 같은 방법을 통하여 조작하는 것을 방지하여야 한다. Wallet 이 이러한 사용자가 사용하는 PChain 을 유지하기 위하여 LWR(Last W Representation)이라는 정보를 유지한다. 이 LWR 은 사용자가 사용한 가장 마지막 체인의 값을 가지고 생성 됨으로써, 사용자가 메모리 조작을 통하여 PChain 의 사용을 방지한다. 그러나 LWR 값도 조작할 가능성이 발생한다. 따라서 Wallet 은 이 LWR 의 Checksum 인 CSR(Current Status Representation)이라는 정보를 사용함으로써 LWR 의 조작여부를 감지할 수 있다.

3.2 Broker 와 사용자와의 관계

사용자는 상거래에 참가하기 위해서는 Broker 로부터 계정정보에 대한 보증서인 Certificate 를 발급 받아야 한다. 사용자는 안전한 경로를 통하여 계정정보에 필요한 정보를 사용자의 공개키로 암호화 하여서 Broker 에게 준다. 계정등록정보를 받은 Broker 는 사용자가 보내온 정보가 정당하면 계정정보에 대한 보증서인 Certificate 를 사용자에게 발급한다.

기존의 Payword 프로토콜에서 Broker 가 사용자에게 Certificate 를 발급할 때, 사용자에게 사용할 PChain 의 루트 값을 전송하지 않았다. 하지만 개선된 모델에서는 사용할 PChain 의 루트 값을 전송함으로써 Vendor 가 대금결제를 요구할 때, Vendor 가 전송하는 사용자의 PChain 의 유효성을 보다 증진시킬 수 있다.

3.3 Vendor 와 사용자와의 관계

사용자가 Broker 로부터 Certificate 를 발급을 받은 후에, 사용자는 Vendor 의 아이템을 구입할 수 있다. 사용자는 Vendor 의 웹 페이지에서 상품을 검색하고 구매할 의사가 있을 때, 구입 대금에 해당하는 PChain 을 전송하기 전에 Commitment 를 전송하여야 한다. 개정된 Payword 프로토콜은 Commitment 를 생성할 때, 사용자가 Broker 와 약정한 금액의 전부에 해당하는 PChain 의 루트 값을 전송하는 것이 아닌, 현재 사용하고 있는 PChain 의 값을 전송한다. 따라서 사용자가 사용할 수 있는 잔액 만큼만 Vendor 에게서 유효한 셈이 된다. 이 Commitment 를 생성할 때 Wallet 은 LWR 값을 생성하게 되는데, LWR 은 다음과 같이 생성된다.

$$LWR = H(W_R, MS, W_{LI})$$

W_R 은 Broker 에서 전송받은 PChain 의 루트 값, MS 는 사용자 계정에 대한 Broker 가 생성한 유일한 값, W_{LI} 는 LI 인덱스에 해당하는 PChain 값이며, LI 는 현재 사용한 PChain 의 루트 값으로부터의 인덱스 값이고, $H(x)$는 단방향 해쉬 함수 이다.

Commitment 가 전송된 후, Wallet 은 사용자가 구입한 아이템의 가격만큼에 해당하는 PChain 을 생성하여 Vendor 에게 전송한다. Wallet 은 PChain 과 더불어 LWR 을 전송한다. Vendor 는 Wallet 이 보낸 LWR 을 기초로 아이템의 가격만큼에 해당하는 LWRU(LWR Update)를 생성한다. LWR 을 전송하는 이유는 Wallet 은 반드시 LWR 값을 사용한 PChain 값 만큼 변경 시켜야 하는데, Vendor 가 보내온 LWRU 와 Wallet 이 전송한 LWR 을 해당 가격만큼으로 다시 계산하여 일치하는 지를 확인하기 위해서 이다.

LWRU 를 받은 Wallet 은 LWR 값을 반드시 LWRU 와 일치하는 값으로 변경시켜야 하고, 변경시킨 LWR 에 해당하는 Checksum 을 계산하여 보관하는데 이를 CSR(Current Status Representation)이라고 한다. 이 CSR 값을 통하여 사용자가 메모리 조작을 통하여 LWR 을 위조했는지 여부를 판별할 수 있다. CSR 은 다음과 같이 생성된다.

$$CSR = H(LWR, LI, H(E_{Wallet}(LWR \odot LI \odot TIME)), TIME)$$

LI 는 현재 사용한 PChain 의 루트 값으로부터의 인덱스 값이고, TIME 은 CSR 값이 생성되는 당시의 시간 값이다. 이 시간값을 사용하여 사용자가 Backward 조작으로부터 CSR 을 방지한다. CSR 은 중간 단계에서 Wallet 의 키로 $LWR \odot LI \odot TIME$ 의 정보를 Wallet 의 키로 암호화 한다. 따라서 사용자가 Wallet 의 키를 사용함으로써 CSR 값을 위조하는 것을 어렵게 하고, Wallet 이 Wallet 의 키를 계속 주기적으로 갱신하여 사용자가 Wallet 의 키에 대한 공격으로부터 Wallet 을 보호한다.

3.4 Vendor 와 Broker 와의 관계

Vendor 는 사용자로부터 모은 PChain 을 Broker 에 전송하여 대금 결제를 요구한다. 사용자가 생성한 PChain 의 루트 값을 Broker 가 계산하여 전송하였으므로, Vendor 가 전송하는 PChain 의 값은 Broker 가 쉽게 계산해 낼 수 있고, Vendor 가 사용자의 체인값을 위조하여 더 많은 금액을 요구하는 것으로부터 사용자를 보호할 수 있다.

4. 결론 및 향후 과제

본 논문에서는 Payword 프로토콜의 단점인 사용자가 악의적인 Payword 체인을 남용하는 것을 방지하는 기법을 Wallet 을 통하여 적용한 새로운 Payword 프로토콜을 제안하였다. 이 프로토콜을 기반으로 Web 을 기반으로 한 소액대금지불시스템을 보다 현실적으로 구현할 수 있다. 또한 사용자의 악의적인 공격으로부터 Vendor 를 보호하고, Broker 가 생성하는 PChain 의 루트 값을 기초로 하여, 보다 사용자와 Vendor 사이의 신뢰성을 높일 수 있다.

제안한 시스템을 기초로 Web 기반의 소액대금지불시스템을 구현하여, 구현 시에 발생하는 문제를 보완하고 이를 개선하여 보다 안전하고 효율적이며 빠른 소액대금지불 프로토콜을 설계하는 것이 향후 과제이다.

5. 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://millicent.digital.com/html/papers/millicent-w3c4/millicent.html

[3] R. Hauser, M.Steiner, and M.Waidner, "Micro-Payments based on iKP", Available from authors, December 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] Ronald 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, August 1995.

[8] Michael Morrison, " JAVA UNLEASHED", Sams Publishing 1996