모바일 전자상거래를 위한 ID 기반 지불 프로토콜
(ID-based Payment Protocol for Mobile Electronic Commerce)

이 현 주 † 김 선 신 †† 이 충 세 †††
(Hyun-Ju Lee) (Sun-Shin Kim) (Chung-Sei Rhee)

요 약 M-commerce가 활성화되기 위한 주요 요건 중의 하나는 안전성과 효율성을 갖춘 전자 지불 시스템을 개발하는 것이다. 본 논문에서는 ID 기반 공개키 암호 시스템을 이용하여 다중 거래에 적용할 수 있는 효율적인 소액 지불 프로토콜 (Micro-Payment Protocol)을 제안한다. 기존의 PayWord 시스템은 다수의 판매자와 거래를 하기 위해 매번 판매자의 인증서를 생성하였다. 본 논문에서는 인증서 대신 유한체 F_q에서 타원곡선(Elliptic Curve Cryptosystem)을 이용한 Weil pairing에 의해 생성된 세션키를 거래에 사용하기 때문에 알려진 키 공격(Known key attacks)과 위장 공격(Man-in-the-middle attacks)에 안전하다.

키워드 : ID 기반 공개키 암호 시스템, 소액 지불, 알려진 키 공격, 위장 공격

Abstract Design an efficient and secure electronic payment system is important for M-Commerce. In this paper, we propose an efficient Micro-Payment Protocol that allows multiple transactions using ID-based public key cryptosystem. Current PayWord system requires to generate certificate of the vendor for each transaction. In this paper, we use a session key instead of certificate key generated by Weil Pairing which use an Elliptic Curve Cryptosystem over finite field F_q for transactions. Therefore, it is more secure in Known key attacks as well as Man-in-the-middle attacks.

Key words : ID-based Public Key Cryptosystem, Micro-Payment, Known key attacks, Man-in-the-middle-attacks

1. 서 론

정보 통신 기술의 발전과 모바일 단말기 사용의 증가로 무선 인터넷 전자 상거래 시장이 급속하게 확산되고 있으며 제공되는 서비스가 다양해질 전망이다[1]. 기존 개인용 컴퓨터 등의 고정 단말기를 기반으로 한 E-Commerce 형태를 벗어나 이제는 이동성(mobility), 휴대성(portability)을 제공하는 새로운 형태의 M-Commerce가 보편화되고 있다. 이러한 M-Commerce에서 안전한 서비스를 제공하기 위해 특성에 알맞은 결제 서비스의 연구가 활발하게 진행되고 있으며 현재 무선 결제 서비스는 소액 결제 서비스가 주로 이루어지고 있다. 미래 무선 전자상거래는 전자 신문, 전자저널, 게임, MP3 같은 음악 화일, MPEG나 AVI같은 동영상 화일 및 GIF나 JPEG같은 그림 화일 등 소액의 정보 상품이 차지하는 비중이 높아질 것이다. 반면, 전자지불 시스템들이 소액 지불에 사용될 경우 상품 가격에 비해 상대적으로 높은 지불 처리 비용을 부담해야 하는 단점을 갖고 있다. 이러한 소액 정보 상품에 대한 전자지불을 효과적으로 하기 위한 전자지불 프로토콜에는 Milli-Cent, PayWord, MicroMint, MPTP 등이 있다[2-4]. 대부분의 소액 지불 프로토콜에서는 MD5와 같이 암호학적으로 강한 일방향 해쉬 함수를 반복해서 수행하는 해쉬 체인 기법을 사용한다[5]. 이는 공개키 암호 연산에 비해 수행 속도가 빠르고 비용이 저렴하기 때문이다. 일반적으로 해쉬 함수는 RSA 방식으로 서명을 수행하는 시간보다 10,000배 정도 빠르고, 이를 검증하는 시간도 100배 정도 빠르다[3]. 따라서 소액 지불 시스템에서는 공개키 암호 알고리즘의 사용을 최소화하고 처리 속도와 비용을 고려하여 해쉬 함수와 같은 암호 알고리즘을 사용하도록 설계되어야 한다. 기존 PayWord 프로토콜은 n번의 거래마다 브로커가 판매자의 인증서를 매번 생성해야 하며 주기적으로 갱신하고 폐지해야 하는 집

이 논문은 2004년도 충북대학교 학술지원 사업의 연구비 지원과 충북대학교 컴퓨터정보통신연구소 지원에 의하여 연구되었음
† 학생회원 : 충북대학교 컴퓨터과학과
pinklee104@korea.com
†† 비 회 원 : 충북대학교 컴퓨터과학과
sskim04@hotmail.com
††† 종신회원 : 충북대학교 컴퓨터과학과 교수
csrhee@cbucc.chungbuk.ac.kr
논문접수 : 2003년 9월 17일
심사완료 : 2004년 3월 30일

중화 문제가 잠재되어 있다. 이런 경우 상품을 주문하여 지불하고 결제하는 과정에서 실시간 응답 시간이 중요한 경우 적합하지 않다. 또한, 브로커가 발행한 사용자와 판매자의 인증서는 두 개체끼리의 상호 인증이 불확실하다[6].

본 논문에서는 M-Commerce 환경에 적합하도록 각 개체의 ID를 사용하여 공개키/개인키를 만들고 Weil pairing에 의해 생성한 세션키를 두 번째 거래부터 판매자의 인증서 대신 사용하여 인증서의 생성 횟수를 줄임으로써 집중화 문제 및 속도를 향상시키고자 한다. 또한 사용자, 판매자, 브로커가 각각 세션키를 생성함으로 역할을 분담하고 사용자와 판매자 사이에 상호 인증이 확인되므로 알려진 키 공격과 위장 공격에 대한 안전성을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고, 3장에서는 ID 기반의 공개키 암호 시스템을 설명한다. 4장에서는 제안하는 ID기반 지불 프로토콜을 기술한다. 5장에서는 제안한 지불 프로토콜의 안전성 및 효율성을 분석하고, 6장에서는 결론을 제시한다.

2. 관련연구

소액 지불시스템은 전자화폐 시스템의 특수한 형태로서, 소량의 금액 지불을 전문적으로 처리하기 위하여 고안되었다. 또한 처리 비용을 최소로 하여 지불 처리에 소요되는 비용이 상품의 가격보다 낮고 지불 금액이 소규모이므로 시스템의 오류로 인한 피해가 다른 지불 시스템보다 적은 장점이 있다. 본 장에서는 전자지불 시스템의 대표적인 프로토콜인 MilliCent, PayWord를 소개한다.

2.1 MilliCent

밀리센트 전자화폐 시스템은 현재 신용카드나 기타 지불 시스템으로 지불 처리를 하기 어려운 소액 거래를 이루기 위해 미국의 DEC(Digital Equipment Corp)사에서 개발한 소액 지불 시스템이다[2]. 이 프로토콜은 크게 사용하는 전자화폐인 스크립(Scrip)과 스크립을 판매하는 브로커를 중심으로 하고, 여기에 물품을 판매하는 상점(Vendor)이 참여하는 가입자 기반의 전자화폐 시스템이다. 그리고 사용자는 클라이언트 소프트웨어인 MilliCent Wallet을 사용하여 물건을 구입하고 대금을 지불한다. 그림 1과 같이 사용자는 브로커에게 Scrip을 받아 상점에게 그 Scrip을 지불함으로써 원하는 상품이나 서비스를 구매할 수 있다. 일반적으로 상점이 직접 Scrip을 발행하지만, 상점이 Scrip을 발행하지 않고 브로커에게 Scrip 발행에 필요한 parameter를 건네주어 대신 발행케 할 수도 있다. MilliCent에서는 물건을 구입하기 위해 현금 대신에 스크립이라고 하는, 각 상점마다 다른 독특한 전자 현금을 사용한다. 스크립은 미리 대금을 지불한 선불 형태이며 특정한 상점에게만 사용이 가능하다.

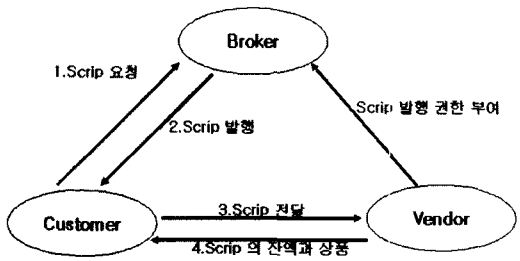

그림 1 MilliCent의 구조

MilliCent 프로토콜은 암호화 알고리즘을 사용하지 않고, 메시지 다이제스트를 이용하여 Scrip을 함으로써 지불 비용을 최소화 하였다. 반면, 이 시스템은 거래 주체들끼리 공유키를 사용하여 Scrip의 유효성 여부를 판단하기 때문에 공유키만을 위한 별도의 데이타베이스를 유지해야 하는 단점을 가지고 있으며, Scrip의 보안에 대한 내용을 Scrip의 발행인만이 알고 있기 때문에 소비자가 브로커로부터 받은 Scrip에 대한 유효성 여부를 판단할 수 없다는 문제점을 가지고 있다. 또한, 거래후의 잔액에 대해서는 상점이 잔액 Scrip을 발행하기 때문에 제한적인 연속 거래가 가능하지만, 마지막 거래까지 잔액이 남는 경우 잔액처리의 부담을 소비자가 안게 되는 단점이 있다.

2.1.1 스크립의 특징

- 특정 상점에게만 사용할 수 있는 화폐 가치를 포함한다.
- 단 한 번만 사용이 가능하다.
- 위조에 대해 보안성을 가진다(인증서를 이용하여 확인)
- 다른 사람이 아닌 정당한 사용자만이 스크립으로 지불할 수가 있다.
- 스크립을 만들어 내고 유효성을 확인하는데 효율적이다(간단한 해쉬 함수만을 사용).

2.1.2 스크립의 구조

스크립은 고유의 값어치를 가지고 있다는 면에서는 현금과 같으나 특정한 판매자에게만 보내어질 때 그 값어치를 지닌다는 면에서는 현금과 차이가 있다. 밀리센트 시스템에서는 브로커 또는 판매자가 스크립을 발행을 할 수가 있다.

그림 2 스크립의 구조

- Vendor: 스크립을 발행한 상거래 서버의 ID
- Value: 스크립의 화폐 가치
- ID#: 스크립의 이중 사용을 막기 위한 스크립의 유일한 번호
- Cust_ID#: 스크립을 사용하는 고객의 ID
- Expires: 스크립의 유효 기간
- Props: 기타 데이타
- Certificate: 스크립에 대한 변조 여부의 확인을 위한 인증서

2.2 PayWord

PayWord 프로토콜은 해쉬 체인을 이용하여 전자화폐인 payword를 사용자가 직접 발행하는 것이 특징이다[3]. 고객은 브로커에게 신용카드 번호를 전송하여 인증서를 발급 받아 payword를 생성한다. 브로커가 서명한 인증서 C_U에는 브로커의 이름 B, 사용자의 이름 U, IP주소 A_U, 고객의 공개키 PK_U, 유효기간 E, 그리고 다른 기타 정보 I_U를 포함한다.

$$C_U = \{B, U, A_U, PK_U, E, I_U\}_{SK_B}$$

인증서는 사용자가 payword 해쉬 체인을 만들 수 있는 권한을 부여한다. 또한 인증서는 일정 기간이 지나면 브로커가 고객의 계좌가 좋은 상태인지를 검사하여 재발급한다.

2.2.1 PayWord 생성

마지막 payword인 w_n을 임의로 정하고, w_n을 제외한 나머지 payword들은 $w_i = h(w_{i+1})$, $(i = n-1, n-2, \ldots, 0)$을 계산함으로써 체인을 생성한다. 그런 다음, 사용자는 그 체인에 대해 아래와 같은 사용자의 개인키 SK_U로 서명한 위임 메시지를 만든다. V는 상인 이름, D는 현재 시간, I_M은 기타 정보를 뜻한다.

$$M = \{V, C_U, w_0, D, I_M\}_{SK_U}$$

payword는 상인마다 유일한 체인을 사용하기 때문에 사용자는 지불할 상인마다 별도의 위임 메시지를 만들어야 한다.

2.2.2 PayWord 지불 과정

사용자의 i번째 지불은 (w_i, i)로 구성되며 상인은 w_{i-1}을 사용한 해쉬 연산으로 유효성을 확인할 수 있다. 하루의 마지막에 상인은 각 사용자에게 받은 마지막 지불인 $P_l = (w_l, l)$과 이에 대응하는 위임 메시지를 함께 브로커에게 보낸다. 브로커는 l번의 해쉬 함수를 반복 수행하여 w_l을 확인한 후, 사용자의 계좌에서 l만큼의 금액을 청구하여 상인의 계좌로 지급한다. payword는 각 상인에 대한 고유한 payword를 사용하기 때문에 상인이 독립적으로 payword의 이중 사용, 위조, 변조를 검사할 수 있다는 장점이 있다. 그러나 PayWord 시스템은 몇 가지 결점을 가지고 있다.

2.2.3 PayWord 프로토콜의 결점

PayWord 프로토콜은 암호 연산 속도가 빠르고 비용이 낮은 해쉬 체인 연산을 사용한다. 이러한 해쉬 함수의 특성은 소액 지불 프로토콜에 적합하므로 대부분의 소액 지불 시스템에서 이를 도입하여 사용하고 있다. 그러나 PayWord 프로토콜은 다음과 같은 몇 가지 결점을 지니고 있다[5].

- 사용자는 거래하고자 하는 상인마다 다른 해쉬 체인을 사용해야 한다. 그러므로 사용자의 측면에서, 상인의 수만큼 공개키 연산을 수행해야 한다.
- 사용자는 신용 한도를 초과하여 화폐를 남용할 수 있다.
- 사용자는 거래에 사용되었던 마지막 인덱스 값을 모두 저장해야 한다.

2.3 그 외 소액 지불 프로토콜

그 외의 소액 지불 프로토콜에는 MITLCS의 Ronald Rivest와 Weizmann Institute of Science의 Adi Shamir가 제안한 MicroMint와 PayWord 방식을 변형한 MPTP, 그리고 PayWord방식과 Ecash 방식을 혼용하여 제안한 Wenbo Payment가 있다[4,7]. MicroMint는 minted 라고 불리는 coin을 브로커가 생성하고 사용자는 브로커로부터 coin을 사서 상점에 지불하는 방식이다. 그러나 소비자의 요구보다 많은 coin을 생산하기 때문에 자원의 낭비가 발생하고, 위조 방지를 위해 Hash 이외의 부가적인 연산을 수행해야 한다. 표 1은 각 지불 프로토콜에 대한 비교이다.

3. ID 기반 공개키 암호 시스템

3.1 ID 기반 공개키 암호 시스템

표 1 소액 지불 시스템의 비교

	Denomination	Produced by	Anonymity	Credit- or Debit-based?	Security
MilliCent	V-or B-specific Scrip	V and/or B	No	Debit to C, V and B	Symmetric encryption, Signature, Hash
MicroMint	Coins satisfying monthly criterion, U_id, and V_id	B	No	Debit to C and B, Credit to V	Signature, k-way Hash
PayWord	U-V-specific chains of PayWords	U	No	Credit	Signature Hash

본래의 PKC(Public Key Cryptosystem)는 공개키를 인증하고 관리하는 인프라 구축에 많은 비용이 든다. 반면, ID 기반 PKC는 이런 문제점을 해결하고 있다. ID 기반 PKC에서 모든 사람의 공개키는 사전에 이메일 주소와 같은 정보에 의해 결정된다. Shamir에 의해 제안된 이 개념은 원래 e-mail 시스템에서 인증 관리를 단순화하기 위한 것이었다[8]. Alice가 Bob에게 bob@hotmail.com으로 메일을 보낼 때 공개키 스트링 bob@hotmail.com을 사용하여 메시지를 암호화한다. Alice는 Bob의 공개키 인증서를 획득할 필요가 없다. Bob은 메시지를 복호화하기 위해 KGC(Key Generation Center)에게 자신을 인증한 후 자신의 개인키를 얻는다. 기존의 e-mail 구조와 달리, Alice는 Bob이 사전에 공개키 인증을 설정하지 않아도 암호화된 메일을 보낼 수 있다. ID 기반 시스템에서는 KGC가 Bob의 개인키를 알고 있을때 key escrow는 고유하다. ID-based 시스템에서는 신뢰할 수 있는 KGC가 필요하다. KGC에서는 각 개체의 ID 기반 공개키를 사용하여 개인키를 생성한다.

3.2 Weil Pairing

Weil pairing은 타원곡선 이산대수 문제의 공격에 사용되어왔으며 3자 키 공유 시스템의 구성도 가능하다[9]. Weil pairing은 초특이 타원곡선 상에서 정의되는 쌍선형사상(bilinear map)이다. G가 유한체 F_q상에서 초특이 타원곡선 위의 점으로 이루어진 군(group)이라 하자. G의 위수(order)를 l로 표기하고, $l|q^k-1$을 만족하는 가장 작은 정수 k를 정의하자. Weil pairing $\hat{e}: G\times G \rightarrow F_q^*$ 는 다음과 같은 성질을 만족한다.

① Bilinear

- $\hat{e}(P_1+P_2, Q) = \hat{e}(P_1, Q) \cdot \hat{e}(P_2, Q)$
- $\hat{e}(P, Q_1+Q_2) = \hat{e}(P, Q_1) \cdot \hat{e}(P, Q_2)$
- $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, $a, b \in Z_q^*$

② Non-Degenerate

$\hat{e}(P, P) \neq 1$ 인 $P \in G$ 가 존재한다.

③ Computable

$\hat{e}(P, Q)$는 다항식 시간(polynomial time)으로 계산할 수 있다.

4. 제안하는 ID기반 지불 프로토콜

본 논문에서는 각 개체의 ID을 사용하여 공개키/개인키(public key/private key)를 생성함으로써 공개키를 인증(authentication)할 필요가 없다. 또한, 유한체 F_q에서 타원곡선(Elliptic Curve Cryptosystem)을 이용한 Weil pairing 에 의해 세션키를 생성하고, ID 기반 3자간의 키 동의 프로토콜을 적용하여 기존의 소액 지불 프로토콜(Micro-Payment Protocols)에 ID기반 암호화를 적용함으로써 속도의 향상 및 뛰어난 안전성을 갖는 새로운 소액 지불 프로토콜을 제안한다. 다음은 기존 프로토콜로부터 새롭게 제안하는 ID기반 지불 프로토콜의 향상된 내용이다.

- 개체의 ID를 사용하여 공개키/ 개인키를 생성한다.
- ECC를 이용한 Weil pairing에 의해 세션키를 생성한다.
- n명의 판매자와 거래시 인증서를 1회 사용한다.
- ID 기반 공개키 알고리즘을 적용하여 안전성을 향상시킨다.

4.1 시스템 설정

본 논문에서 브로커는 ID 기반 시스템에서 KGC 역할을 한다고 가정한다. 사용자는 브로커에게 자신의 인증서 암호화에 필요한 공개키 생성 요청을 위해 안전한 채널로 자신의 ID를 전송한다.

- $H: F \rightarrow 0,1^*$: 키 유도 함수(key derivation function)
- $H: 0,1^* \rightarrow G$: 해쉬함수(hash function)

표 2 시스템 설정에 필요한 파라메터

U, V, B	사용자, 판매자, 브로커
Z	$Z \in U, V, B$
Z_{ID}	Z의 ID
W_Z	Z의 공개키
w_Z	Z의 개인키
k_Z	Z의 세션키
C_Z	Z의 인증서
D_Z	Z로 가장한 적(adversary)

표 2는 공개키/개인키, 세션키 생성에 필요한 파라메터를 나타낸다. 브로커는 비밀키 $s \in 1, \ldots, l-1$와 난수 $P \in G$을 선택한 후, $P_B = [s]P$를 계산한다. 그리고 (P, P_B)는 공개한다. 사용자, 판매자, 그리고 브로커는 세션키를 공유하길 원한다고 정의한다. 사용자는 브로커에게 자신의 아이디를 보낸다. 브로커는 사용자의 공개키 $W_U = H(U_{ID})$를 생성하고 개인키 $w_U = [s]W_U$를 생성한다. 판매자의 공개키/개인키도 같은 방법으로 브로커에 의해 생성된다. 두 번째 거래부터 판매자의 인증서 대신 세션키 k_{UVB}를 사용하여 거래가 이루어진다. 사용자, 판매자, 브로커는 각각 개인키 역할을 하는 short-term key인 난수 $a, b, c \in Z_q^*$를 생성한다. 세션키 생성 프로토콜은 다음과 같다.

• $U \rightarrow V$: $[a]P, [a]W_B$; $U \rightarrow B$: $[a]P, [a]W_V$

• $V \rightarrow U$: $[b]P, [b]W_B$; $V \rightarrow B$: $[b]P, [b]W_U$

• $B \rightarrow U$: $[c]P, [c]W_V$; $B \rightarrow V$: $[c]P, [c]W_U$

사용자, 판매자, 그리고 브로커는 세션키를 계산한다.

$$k_U = \hat{e}([a](W_V + W_B), P_B) \cdot \hat{e}(w_U, ([b]P + [c]P)) \cdot \hat{e}([b]W_B, P_B) \cdot \hat{e}([c]W_V, P_B)$$

$$k_V = \hat{e}([b](W_U + W_B), P_B) \cdot \hat{e}(w_V, ([a]P + [c]P)) \cdot \hat{e}([a]W_B, P_B) \cdot \hat{e}([c]W_U, P_B)$$

$$k_B = \hat{e}([c](W_U + W_V), P_B) \cdot \hat{e}(w_B, ([a]P + [b]P)) \cdot \hat{e}([a]W_V, P_B) \cdot \hat{e}([b]W_U, P_B)$$

따라서, 공통 세션키는 키유도함수(key derivation function) H의 값이 된다.

$$k_{UVB} = k_U = k_V = k_B$$
$$= \hat{e}([a](W_V + W_B) + [b](W_U + W_B) + [c](W_U + W_V), [s]P)$$

long-term secret keys를 사용하여 생성한 세션키는 새 개체의 W_U, W_V, W_B와 KGC의 비밀키 s, 그리고 개인키 a, b, c에 의해 결정된다.

4.1.1 세션키의 안전성

3자간의 키 동의 프로토콜(key agreement protocol)에 의해 생성된 세션키는 다음과 같은 안전성을 제공한다[8].

- Forward secrecy: long-term key(s)가 유출되어도 이전에 사용한 세션키를 유출하지 않는다.
- Key Independence: A그룹의 키를 아는 수동적인 공격자가 다른 B그룹의 키를 알아낼 수 없기 때문에 키 독립성을 제공한다.
- Key Freshness: 적(adversary)에게 이전키(old key)가 재사용되어도 새로 생성된 키는 key freshness를 제공한다.
- Known Key attacks: 이전 세션키가 유출되었을 때 수동적인 적이 다음 세션키를 알 수 없고, 능동적인 적이 프로토콜에 참여한 한 개체를 가장하여 다음 세션키를 알아낼 수 없기 때문에 알려진 키 공격에 안전하다.

4.2 ID 기반 소액 지불 프로토콜

사용자가 첫 번째 판매자와 거래 하는 경우와 k번째 판매자와 거래하는 경우의 지불 프로토콜을 제안한다. 표 3은 상품 거래 과정에서 사용되는 알고리즘이다.

표 3 지불 프로토콜에 사용되는 알고리즘

알고리즘	설 명
$h()$	충돌 회피 해쉬 함수
$Sign_Z data$	Z의 개인키를 사용하여 메시지 서명

4.2.1 $1th$ 판매자와의 지불 프로토콜

● 인증서 획득 단계

1) 사용자는 인증서 획득을 위해 브로커와 미리 설정된 안전한 통신 채널을 통해 해쉬 체인의 root값 w_0, 해쉬 체인의 길이 n, 사용자의 아이디 U_{ID}, 브로커의 아이디 B_{ID}를 포함한 메시지를 브로커의 공개키로 암호화하여 전송한다.

$$U \rightarrow B: w_0, n, U_{ID}, B_{ID}{}_{W_B} \quad (1)$$

2) 브로커는 받은 메시지를 개인키로 복호화한 후 해쉬 체인의 길이가 사용자의 계정에서 사용 가능한지를 체크한다. 해쉬 체인의 길이가 초과되지 않으면 브로커는 사용자에게 인증서의 유효기간 E를 포함한 인증서를 발급한다.

$$B \rightarrow U: C_U = Sign_B w_0, n, U_{ID}, B_{ID}, E \quad (2)$$

브로커가 서명한 인증서는 정당한 사용자에게 해쉬 체인의 생성 권한을 부여한다. 사용자는 다음과 같은 경우에 해쉬 체인을 생성한다.

- 해당 판매자에 관한 해쉬 체인을 모두 소비하였을 경우
- 해쉬 체인(또는 인증서)의 유효기간이 지났을 경우

3) 또한, 판매자는 사용자에게 거래 영수증을 발급할 때 사용될 인증서를 브로커에게 받는다. 판매자의 인증서 C_V에는 판매자의 아이디 V_{ID}, 브로커의 아이디 B_{ID}, 인증서 유효기간 E를 포함한다.

$$B \rightarrow V: C_V = Sign_B V_{ID}, B_{ID}, E \quad (3)$$

● 상품 요청 및 지불 단계

사용자는 인터넷을 통해 구입하고자 하는 상품의 정보를 찾는다. 사용자와 판매자 사이의 거래는 사전에 정해진 시간 내에 이루어져야 한다.

1) 사용자는 판매자의 웹 사이트로부터 상품 구매 요청을 위해, C_U, 판매자의 서명이 되어 있는 상품 아이디 $ProductID$, 구매자의 신원과 거래 확인을 위한 $Sign_U(k_U)$, 거래 시간 t 등의 메시지를 보낸다. 또한, 사용자는 주문한 상품과 다른 상품을 받았을 때 $ProductID$를 통해 확인한다.

$$U \rightarrow V: Product\ request \quad \{V_{ID}, U_{ID}, C_U, ProductID, Price, t, Sign_U(k_U)\}_{W_V} \quad (4)$$

2) 판매자는 인증서의 만료 기간을 확인하고 인증서로부터 해쉬 체인의 root 값, 해쉬 체인의 길이 등을 검증한다. 확인 후, 판매자는 상품을 대칭키 K로 암호화하고, k_V에 서명한 값과 상품의 $Price$를 사용자의 공개키로 암호화하여 보낸다.

$$V \rightarrow U: Goods\ Delivery \quad [goods]_K \quad \{h[goods]_K, Sign_V(k_V), Price\}_{W_U} \quad (5)$$

3) 암호화된 상품을 전달받은 구매자는 지불을 위한 해쉬 체인 값 w_i과 인덱스 i를 보낸다.

$$U \rightarrow V: Payment = (w_i, i) \quad (6)$$

4) 판매자는 전송받은 해쉬 체인의 길이를 계산하여 root값과 비교 검증한다. 지불 금액 검증 확인 후 판매자는 사용자가 상품을 복호화 할 수 있도록 복호화키 K, 남은 인덱스 길이 $n-i$, C_V, w_i와 C_V, $n-i$에 서명한 영수증을 사용자에게 전송한다.

$$V \rightarrow U: Receipt \\ \{K, n-i, C_V, w_i, Sign_V\{h(C_V, n-i)\}\}_{w_U} \quad (7)$$

5) 영수증을 받은 사용자는 남은 인덱스 길이 $n-i$과 판매자의 인증서 C_V를 확인한 후 K로 상품을 복호화 하여 구매한 상품을 받게 된다.

● 결제 단계

판매자는 사용자에게 영수증을 발급한 후 일정 시간 안에 브로커와 결제 단계를 수행한다.

1) 판매자는 브로커에게 C_U, 해쉬 체인 인덱스와 판매자가 생성한 세션키 k_V에 서명하여 결제를 요구한다.

$$V \rightarrow B: Deposit\ request \\ \{C_U, k_V, w_i, i, Sign_V(k_V)\}_{w_B} \quad (8)$$

2) 브로커는 사용자의 인증서에서 결제 가능한 해쉬 체인의 길이를 확인한다. 브로커는 해쉬 체인 값을 검증한 후, root 값이 일치하면 판매자의 계좌로 결제 금액을 입금시킨다.

$$B \rightarrow V: Redemption \quad (9)$$

4.2.2 k번째 판매자와의 지불 프로토콜

사용자가 k번째 판매자에게 지불하는 경우를 제안한다. 두 번째 거래부터는 인증서 대신 k_{UVB}를 사용하여 인증한 후 거래가 이루어진다. 다음을 가정한다.

- 사용자는 해쉬 체인 길이 n을 갖는다.
- $(k-1)th$ 판매자는 인덱스 i를 갖는다.
- k번째 판매자는 인덱스 j를 갖는다.

1) 상품 요청 단계는 첫 번째 판매자와의 거래 과정과 같다.

$$U \rightarrow V_k: Product\ request \\ \{V_{k_m}, U_{ID}, C_U, ProductID, Price, t, Sign_U(k_U)\}_{w_{V_1}} \quad (10)$$

2) 요청을 받은 판매자는 인증 후 요청한 상품을 보낸다. 대칭키 K로 암호화한 상품, 정당한 판매자의 신원을 위해 k_{V_k}에 자신의 전자 서명, 그리고 사용자에게 지불을 받기 위해 $Price$를 사용자의 공개키로 암호화하여 전송한다.

$$V_k \rightarrow U: Goods\ Delivery \\ [goods]_K \\ \{h[goods], Sign_{V_k}(k_{V_k}), Price\}_{w_U} \quad (11)$$

3) 사용자는 개인키로 복호화하여 자신이 요청한 상품의 가격을 확인한 후 판매자에게 $payment=(w_j, j)$에 해쉬 함수를 수행하여 지불 금액을 확인할 수 있도록 w_{i+j}, j, $n-i$와 $k_{V_{k-1}}$, 그리고 $(k-1)$번째 판매자와 거래 후 남은 해쉬 함수의 길이 $n-i$과 $k_{V_{k-1}}$에 서명한 메시지를 보낸다. 또한, V_k가 브로커에게 결제 요청 시 결제 정보를 위조하지 못하도록 C_U, w_{i+j}, j에 자신의 서명을 한 메시지를 보낸다.

$$U \rightarrow V_k: payment = (w_j, j) \\ \{w_{i+j}, j, n-i, k_{V_{k-1}}, Sign_U\{h(k_{V_{k-1}}, n-i)\}, \\ Sign_U\{h(C_U, w_{i+j}, j)\}\}_{w_{V_k}} \quad (12)$$

4) V_k는 자신의 개인키로 복호화한 후 w_{i+j}에 해쉬 함수를 j번 수행하여 $(k-1)$번째 판매자의 마지막 payword값 (w_i, i)와 같은지를 확인한다. 값이 일치하면 V_k는 사용자에게 다음 거래에 사용가능한 해쉬 체인의 길이 $n-i-j$, 마지막 payword값 w_{i+j}, 상품 복호화키 K와 k_{V_k}의 메시지를 갖는 영수증을 보낸다.

$$V_k \rightarrow U: Receipt \\ \{K, n-i-j, w_{i+j}, k_{V_k}, Sign_{V_k}\{h(k_{V_k}, n-i-j)\}\}_{w_U} \quad (13)$$

5) 판매자는 상품에 상응하는 값을 결제 받기 위해 브로커에게 w_i, w_{i+j}, j와 사용자에게 전송받은 서명된 메시지 $Sign_U h(C_U, w_{i+j}, j)$를 보낸다.

$$V_k \rightarrow B: Deposit\ Request \\ \{Sign_{V_k}(k_{V_k}, n-i-j), Sign_U\{h(C_U, w_{i+j}, j)\}, \\ C_U, w_i, w_{i+j}, j\}_{w_B} \quad (14)$$

6) 브로커는 C_U에 수행된 자신의 서명을 검증하고 판매자가 요청한 금액을 판매자의 계좌로 이체시킨다. 브로커는 w_i에서 w_{i+j}까지만 검증할 수 있다. 즉, root 값은 w_i가 되고 마지막 받은 w_{i+j}에 해쉬 함수를 j번 적용하여 w_i와 같은지만 확인하면 된다. 브로커는 만료 기한까지 다수의 판매자에 대한 결제 요청을 처리하고, 마지막 받은 $payment$가 사용자가 생성한 해쉬 체인의 최대 값인 w_n보다 작은지를 확인함으로써 결제 과정을 완료한다.

$$B \rightarrow V_k: Redemption \quad (15)$$

※ $n=10$, $i=3$, $j=4$ 일 때 해쉬 함수를 적용하여 보자 $(w_{i-1}=h(w_i))$.

① $(k-1)$번째 판매자의 인덱스 $i=3$일 경우

(풀이) $payment=(w_3, 3)$이므로 해쉬 함수를 3번 수행한다.

$$w_2 = h(w_3), \quad i=3 \\ w_1 = h(w_2), \quad i=2 \\ w_0 = h(w_1), \quad i=1$$

해쉬 함수를 3번 수행하여 마지막 인덱스에 대한 해쉬 값 $w_0=h(w_1)$과 이전의 거래에서 전달 받은 root값

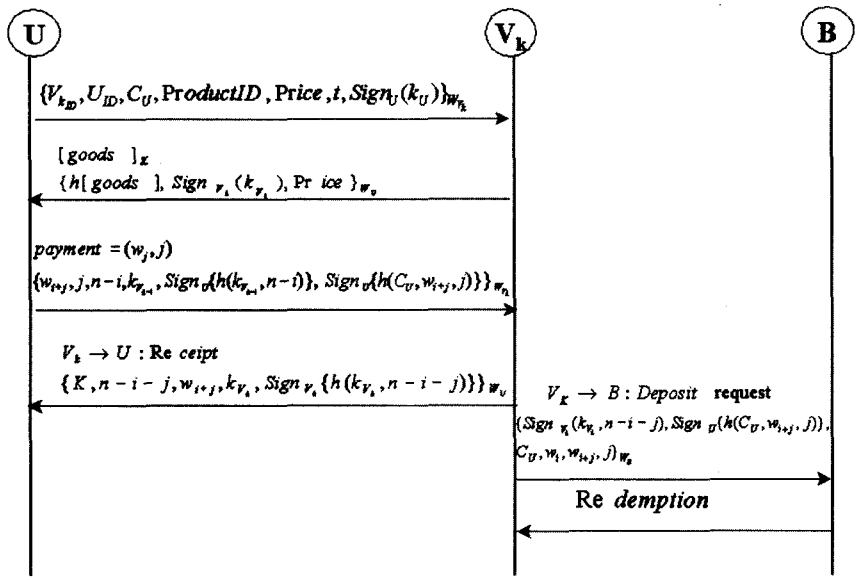

그림 3 k번째 판매자와의 지불 프로토콜

w_0와 같은지를 검증한다.

② k번째 판매자의 인덱스 j=4일 경우

(풀이) $payment=(w_{i+j}, i+j)$이므로 해쉬 함수를 4번 수행한다.

$w_6=h(w_7)$, $i+j=7$
$w_5=h(w_6)$, $i+j=6$
$w_4=h(w_5)$, $i+j=5$
$w_3=h(w_4)$, $i+j=4$

해쉬 함수를 4번 적용하여 마지막 인덱스에 대한 해쉬 값 $w_3=h(w_4)$와 이전의 거래에서 k번째 거래에 root값으로 사용되는 $(w_3, 3)$과 같은지 확인한다.

5. 안전성 및 효율성 분석

소액 지불 시스템에서는 화폐 단위가 낮은 반면 거래가 빈번하기 때문에 모바일 환경에서 지불 데이타를 전송할 때 알고리즘 연산량과 안전성에 대해 고려해야 한다. 표 4, 표 5는 기존 방식과 제안 방식을 비교 분석한 것이다.

5.1 안전성 분석

● 위조 방지

브로커는 다른 사용자가 획득할 수 없는 자신의 개인키로 서명한 인증서를 안전한 채널로 사용자에게 전송하기 때문에 C_U의 정당한 소유자만이 화폐 가치를 지닌 payword를 생성할 수 있다. 따라서, 인증서를 소유하고 있는 사용자만이 전자 화폐를 생성할 수 있는 권한을 갖기 때문에 전자 화폐에 대한 위조가 불가능하다. 또한, 판매자가 결제 정보를 위조하여 브로커에게 지불 정보를 전송하려는 경우, 식 (14)에서 C_U에 서명된 브로커의 비밀 서명키와 사용자의 비밀 서명키를 알아야 하는데 사실상 이는 불가능하다.

● 사용자의 이중지불 탐지

사용자는 상품 요청을 위해 판매자에게 식 (10)을 보낸다. 이때, 사용자의 인증서 C_U에는 해쉬 체인의 길이와 root값을 가지고 있다. 상품을 받은 후 사용자는 판매자에게 *payword*를 지불하기 위해 식 (12)를 수행한다. $payment=(w_j, j)$를 전달 받은 판매자는 w_{i+j}에 해쉬 함수를 j번 적용하여 이전 거래에서의 root값 w_i와 같은지를 확인한다. 즉, $w_{i+j-1}=h(w_{i+j})$와 (w_i, i)가 일치하는지 확인하면 브로커가 *payword*의 이중 지불을 탐지하기 전에 판매자가 미리 알 수 있다. 또한, 사용자가 동일한 *payment*를 k와 $k+1$번째 판매자에게도 전송한다면 브로커가 이중 지불을 탐지할 수 있다.

● 판매자의 이중지불 탐지

사용자가 식 (10)을 가지고 이중 지불을 한다면 브로커는 식 (15)에서 판매자의 거래 은행을 알고 있기 때문에 판매자의 거래 은행과 다르다는 것으로 사용자의 이중 지불을 탐지할 수 있다. 그러나 브로커가 사용자에게 계좌 이체를 한다 해도 결국 사용자의 계좌에서 이체가 되기 때문에 사용자가 이중 지불을 요청할 이유는 없다. 따라서 이러한 경우에는 판매자가 식 (10)을 가지고 이중 지불을 하려는 경우이다.

k번째 판매자가 자신이 획득한 *payment*를 이용하여 자신이 사용자 U로 가장하여 다른 판매자에게 동일한

*payment*를 지불하려는 경우 즉, 식 (10)에서 U'(사용자로 가장한 k번째 판매자)는 $\{V_{k_{ID}}, U_{ID}, C_U, ProductID, Price, t, Sign_U(k_U)\}_{W_V}$에서 사용자의 ID와 전자서명을 이용하여 이중 지불을 원할 경우 이때, 은행의 데이타베이스에는 이전 거래에서 사용된 해쉬 체인의 루트 값이 저장되어 있기 때문에 판매자의 이중 지불을 탐지하게 된다.

● 금액 초과 지불 방지

브로커가 발급한 사용자의 인증서에는 해쉬 체인의 최대 크기 n과 같은 지불 정보가 들어 있다. 사용자가 판매자에게 상품 주문 시 인증서 C_U와 지불 정보가 전송되므로 해쉬 체인의 한도를 판매자가 검증할 수 있다. 따라서 사용자가 금액을 초과하여 지불하는 것을 방지할 수 있다.

● 위장 공격 방지

적(adversary) D가 U와 통신하는 V, B의 거래 내역을 도청하는 경우를 가정한다. $\delta, \delta', \delta''$는 각각 D_U, D_V 와 D_B가 생성한 난수이고 세션키를 생성하기 위해 다음과 같은 short-term secret keys 프로토콜을 사용하고 U가 먼저 거래를 하기 위한 통신을 설정한다고 가정한다.

- $U \rightarrow V, B$: $[a]P$
- $V \rightarrow U, B$: $[b]P$
- $B \rightarrow V, U$: $[c]P$

다음과 같은 위장 공격이 이루어진다.

1. $D_{V,B}$는 U로부터 aP를 도청하고, D_U는 V, B에게 δP를 전송한다.
2. D_U는 V로부터 bP를 도청하고, D_V는 U에게 $\delta' P$를 전송한다.
3. D_U는 B로부터 cP를 도청하고, D_B는 U에게 $\delta'' P$를 전송한다.

결국, 이러한 공격은 D_U가 세션키 $k_{D_UVB} = \hat{e}([b]P, [c]P)^{\delta} = \hat{e}(P, P)^{\delta bc}$를 생성하고, $D_{V,B}$는 세션키 $k_{UD_{V,B}}$를 생성한다. 만약 이렇게 생성된 세션키를 가지고 메시지를 복호화한 후 다시 암호화한 후 거래에 사용한다면 D는 U로 위장하여 V, B와 거래를 할 수 있고, V와 B로 위장하여 U와 거래를 할 수 있다. 그러나 제안한 세션키 생성 프로토콜은 k_{UVB}의 계산에서 long-term secret keys를 사용했기 때문에 이러한 위장 공격에 안전하다.

● 알려진 키 공격 방지

이전에 사용된 세션키의 유출로 다음에 사용할 세션키를 적이 생성할 수 있다면 알려진 키 공격에 취약하다. 제안한 프로토콜에서는 long-term secret keys를 사용하였기 때문에 유출된 세션키로 다음 거래에 사용될 세션키를 적이 계산한다는 것은 불가능하다.

표 4 안전성 비교

프로토콜 / 요구사항	PayWord 프로토콜	제안한 프로토콜
위조방지	✓	✓
이중지불탐지	✓	✓
금액초과지불방지	✓	✓
상호인증	✓	✓
위장공격방지	✗	✓
알려진 키공격 방지	✗	✓

✓: 제공, ✗ : 제공하지 않음

5.2 효율성 분석

최근 정보통신망의 급성장과 인터넷의 활용 증대로 국내 통신 시장의 무게 중심이 유선통신에서 유/무선 통합 서비스 시대로 전환이 되고 있다. 즉 기업들은 이런 환경에 적합한 전자상거래, 전자 결제등의 다양한 서비스를 무선 PKI환경에서도 제공하기 위해 많은 노력을 하고 있다. 그러나 무선 환경은 낮은 CPU의 성능, 메모리 용량의 제약, 통신 속도의 둔화, 통신 에러 발생률의 증가 등 많은 제약 조건을 가지고 있어 유선 환경에서의 계산 능력에 미치지 못하고 있다. 그러므로 현재까지 소개된 지불 프로토콜은 좁은 대역폭을 가지고 메모리와 연산 능력도 부족한 무선 환경에 적용하기에는 여러 가지 어려움이 따른다. 본 논문에서 제안한 방식은 무선 환경에 적합하도록 암호 방식의 효율성을 결정하는 계산량을 줄이기 위해 ID기반의 암호화 알고리즘을 적용하였다. ID기반 암호화는 기존의 PKI와 달리 공개키 인증서가 필요 없다. 또한, 타원곡선 알고리즘은 하드웨어 이식이 쉬워 스마트카드나 이동단말기나 호출기와 같이 휴대형 시스템에 적용하기 쉽고 타원곡선에서의 160비트는 RSA에서 1024비트와 같은 안전성을 가지며 주요 연산이 스칼라 곱이기 때문에 수행 시간이 많이 절약될 수 있다.

기존의 PayWord 프로토콜의 경우 사용자가 여러 판매자와 거래를 할 때마다 판매자의 인증서 C_V가 사용되었다. 브로커는 판매자의 인증서를 생성하기 위해 공개키 암호 연산을 매번 수행해야 되기 때문에 공개키 연산량이 빈번해진다. 제안한 ID기반 지불 프로토콜에서는 첫 거래에만 판매자의 인증서를 사용하고 두 번째 거래부터는 각 개체가 ID 기반 공개키 암호에 의한 타원곡선 암호 기법을 적용하여 생성한 세션키로 인증한 후 거래를 시작한다.

표 5는 기존과 제안한 ID기반 지불 프로토콜에서 판매자의 인증서 생성 횟수를 나타낸다. 기존 프로토콜에서는 첫 거래에서 판매자의 인증서가 사용되고 두 번째 거래부터는 전 단계에서 사용된 판매자의 인증서와 거

표 5 알고리즘의 효율성 비교

프로토콜 / 요구사항	PayWord 프로토콜	제안한 ID 기반 프로토콜
n명의 판매자와 거래시 C_V의 생성 횟수	n	1
키생성 알고리즘	공개키 암호	ID-기반 공개키 암호
상품의 암/복호화 키	DES	DES

래하고자 하는 판매자의 인증서가 모두 사용되므로 n명의 판매자와 거래 시 C_V의 생성 횟수는 n 번이고 제안한 ID기반 지불 프로토콜에서는 첫 거래에만 판매자의 인증서가 1회 사용된다. 또한 세션키의 생성은 ID 기반 공개키 암호 알고리즘을 적용했기 때문에 기존 프로토콜에서 공개키 암호 알고리즘을 적용했을 때보다 매우 안전성이 뛰어나고 효율적이다.

6. 결 론

인터넷 통신 기술의 발전과 다양한 형태의 단말기 사용의 증가로 사용자들은 언제 어디서든 필요한 시점에서 모든 서비스를 제공 받을 수 있게 되었다. 그러나 이러한 편리성을 가지고 있는 반면, 보안에 취약하고 공개키 알고리즘의 연산으로 수행 속도와 계산량의 문제를 가지고 있다.

본 논문에서는 기존에 제시되었던 PayWord 프로토콜의 문제점을 분석하여 새로운 ID기반 지불 프로토콜을 제시하였다. 제안한 프로토콜은 ID 기반 공개키 암호 시스템의 적용으로 효율성 및 세션키의 분실이나 오용 등에 의해 발생하는 문제점을 해결할 수 있기 때문에 기존에 제안된 프로토콜에 비해 안전성이 매우 뛰어나다. 향후 모바일 단말기를 통해 지급 결제를 하는 M-Payment는 다양한 장점을 보유하고 있어 차세대 지불 수단으로 주목 받고 있으며 전자화폐, 신분증, 의료카드, SIM 카드 등에 사용되는 스마트카드는 복제가 불가능하기 때문에 높은 보안성을 제공한다. 여기에 ID기반 타원곡선 알고리즘을 적용한다면 보안성 및 속도 향상에서도 효율성을 높일 수 있다. 또한, 지불 과정을 수행하기 전에 은행으로부터 전자화폐를 발급받을 때 은닉서명 기법에 ID기반 암호 알고리즘을 적용한다면 무선 PKI환경에 속도와 안전성 측면에서 효율적으로 적용할 수 있다.

참 고 문 헌

[1] Lyytinen, K., "M-commerce - mobile commerce: a new frontier for E-business," System Sciences, Proceedings of the 34th Annual Hawaii International Conference on, pp.3509-3509, 2001.

[2] Steve Glassman, Mark Manasse, Martin Abadi, Paul Gauthier, and Patrick Sobalvarro, "The MilliCent Protocol for Inexpensive Electronic Commerce," World Wide Web Journal, 1(1), p.89, December 1995.

[3] R.L.Rivest and A.Shamir, "PayWord and MicroMint: Two simple micropayment schemes," CryptoBytes, pp.7-11, 1996.

[4] Phillip M. Hallam-Baker, "Micro Payment Transfer Protocol(MPTP) Version 0.1," W3C Working Draft, 1995.

[5] R.Rivest, "The MD5 Message-Digest Algorithm," Internet RFC 1321, April 1992.

[6] M.H.Lee and K.G.Kim, "A Micro-payment System for Multiple-Shopping," SCIS 2002, Vol.1/2, pp.229-234, January-February 2002.

[7] Ellis CHI, "Evaluation of Micropayment Schemes," HP Lab, technical report, 1997.

[8] Divya Nalla, and K.C.Reddy, "ID-based tripartite Authenticated Key Agreement Protocols from pairings, Cryptology ePrint Archive, Report 2003/004, available at http://eprint.iacr.org/2003/004/.

[9] N.P.Smart, "An Identity based authenticated Key Agreement Protocol based on the Weil pairing," Cryptology ePrint Archive, Report 2001/111, 2001. http://eprint.iacr.org/.

이 현 주

1990년 2월 청주대학교 수학교육과 졸업(이학사). 1992년 2월 청주대학교 수학과 졸업(이학석사). 2000년 8월 청주대학교 수학과 졸업(이학박사). 2003년 2월 충북대학교 대학원 컴퓨터과학과 박사과정 수료. 관심분야는 정보보안, 스마트카드, 전자지불

김 선 신

1989년 2월 충북대학교 물리학과 졸업 1995년 2월 충북대학교 물리학과 석사 2002년 5월 Syracuse University computer science 석사. 2003년 3월~현재 충북대학교 컴퓨터과학과 박사과정. 관심분야는 Bioinformatics, 알고리즘

이 충 세

1989년 University of South Carolina, 전산학 박사. University of North Dakota 전산학과 조교수. 1991년~현재 충북대학교 전기전자 및 컴퓨터공학부 교수. 관심분야는 결함허용, 알고리즘 및 전문가 시스템, 정보보안