이중 계층 은행 구조를 이용하는 온라인 전자현금 시스템의 설계 및 구현

진 성기 홍 성민 윤 현수
한 국 과 학 기 술 원 전 산 학 과
{skjean,smhong,hyoon}@calab.kaist.ac.kr

Design and Implementation of Online Electronic Cash System using 2-Level Bank Structure

Sungkee Jean Seong-Min Hong Hyunsoo Yoon
Department of Computer Science,
Korea Advanced Institute of Science and Technology

요 약

본 논문에서는 온라인 전자현금 시스템을 설계하고 개발한다. 개발된 전자현금 시스템은 공개키 알고리즘, 맹목 서명, cut-and-choose등과 같은 다양한 암호화 기법들이 사용되어 안전성을 높였으며 특히, 전자현금의 이중 사용(double spending)의 효과적인 방지를 위해 2중 은행 구조(2-level bank structure)를 취함으로써 통신망을 오가는 트래픽의 양을 많이 줄이고자 하였다. 이 구조의 가장 큰 특징은 각 은행마다 그 은행에서 발행하는 전자현금을 구분하기 위해서 전자 현금의 정보에 들어 있는 일련번호(sequence number)의 값을 은행별로 범주화한다는 점과, 이를 지원하기 위해서 각 은행마다 Cash Clearing Agent라는 모듈을 가진다는 점이다.

1 서론

컴퓨터가 널리 보급되고, 컴퓨터 네트웍이 커지면서, 사회활동의 많은 부분들이 이들을 이용하여 이루어짐으로써 큰 경제적인 효과를 보고있다. 그러나, 정작 중요한 문제들에 대해서는 보안상의 문제로 인해 기존의 비효율적인 시스템이 이용되고 있으며, 따라서 컴퓨터와 컴퓨터 네트웍 관련 기술들은 보조적인 도움을 주는 수준으로밖에 이용되지 않고 있다. 그러한, 예들로는 전자결재 시스템, 전자지불 시스템, 전자투표 시스템 등이 있다. 그 중에서도 직접 돈을 다루는 전자지불 시스템을 실생활에 이용하기 위해서는 많은 어려움이 따른다. 하지만, 전자지불 시스템이 성공적으로 가동되기 시작하면 우리생활의 많은 부분이 편리해 질 것은 물론이고, 기존의 시스템에서는 기대하기 힘들었던 특성까지 얻을 수 있다.

전자지불 시스템이란 컴퓨터와 컴퓨터 네트웍을 통해서 물품 또는 정보의 대금을 결재할 수 있도록 하는 시스템을 말한다. 이러한 전자지불 시스템의 종류는 기존의 실세계에서의 지불방식에 따라 여러가지로 나누어 볼 수 있다. **신용카드 기반의 전자지불 시스템, 전자현금을 이용한 지불시스템**, 그리고 **전자수표 기반의 지불 시스템**등이 그것들이다.

전자현금은 실생활에서 현금처럼 사용되어야 하기 때문에 사용되는 시점에 전자현금의 유효성을 검증할 수 있어야 한다. 기존에 제시된 전자현금 시스템은 각 트랜잭션에 대한 로그를 남겨 사용 이후에 현금의 유효성을 검증하는 오프라인 시스템이거나 사용중에 유효성을 검증할 수 있는 온라인 시스템이다. 오프라인 시스템의 경우, 그 동작 원리로 인해서 이중 사용이나 비유효한 전자현금의 사용이 트랜잭션 이후에 발견되고, 이때 사용자는 블랙 리스트에 오르며 다음 전자현금의 사용에 있어서 제약을 받게 된다. 온라인 시스템의 경우에는 사용되는 전자현금의 유효성이 트랜잭션 시점에 검증되나 검증을 위한 중앙 은행과의 통신량 증가로 인해서 비효율적인 시스템이 된다.

본 논문에서는 이중 은행 구조를 이용하여 통신망에서의 트래픽을 줄일 수 있는 효율적인 온라인 전자현금 (EOEC : Efficient Online Electronic Cash) 시스템을 설계하고 구현한다. EOEC에서는 각 은행이 발행한 전자현금에 대한 정보를 전자현금 발행 은행이 유지하고, 필요한 경우에만 중앙 은행으로 전송함으로써 전자현금의 유효성을 검증하기 위해 통신망을 오가는 트래픽의 양을 줄일 수 있었다.

본 논문의 구성은 다음과 같다. 2절에서는 기존의 전자지불 시스템들을 종류별로 간략하게 살펴본다. 제 3절에서는 특히 전자현금 시스템의 특징들에 대해서 간략히 살펴보고 4절에서는 본 논문에서 설계하고 구현한 전자현금 시스템인 EOEC에 대한 상세한 설명을 명시한다. 마지막으로 제 5절에서는 결론과 향후 연구과제를 기술한다.

2 관련연구

본 절에서는 기존의 전자지불 시스템에 대해서 간략히 살펴 보기로 한다. 전자지불 시스템에는 신용카드 기반, 전

자현금 기반, 그리고 전자수표 기반등의 지불 시스템들이 있다.

2.1 신용카드 기반의 전자지불 시스템

신용카드는 현재 널리 사용되고 있으며 또한 통신망을 통한 대금 결제 시스템으로 적용하기에 적합하다 [1, 2]. 신용카드 시스템을 이용하는 전자결재 시스템은 크게 다음과 같은 세가지 모델로 나눌 수 있다.

- 평문 방식(Plain Method)
- 암호화 방식(Encryption Method)
- 제 3자 개입 방식(Third Party Processing Method)

평문 방식의 경우, 물품 구입시 신용카드에 관계된 정보를 그냥 평문상태로 상인에게 전달한다. 이 방법은 안전성 측면에서 매우 취약하다. 즉, 개인의 정보도 보호되지 않으며, 상인측에서도 이 정보를 신뢰할 수 없게 된다. 정보보호에 대한 부가적인 연산이 전혀 없기 때문에 다른 방법에 비해서 가격과 효율성 측면에서는 뛰어나다고 볼 수 있지만 안전성에 대한 대책이 전혀 없기 때문에 실제로 사용할 수 없는 시스템이다. 암호화 방식의 경우는 물품 구입시 신용카드에 관계된 정보를 악의의 제 3자에게 밝혀지지 않게 하기 위해 암호화해서 보낸다. 그러나, 중요한 정보들이 암호화되기 이전에 노출될 수 있으며 이 정보들이 악의의 제 3자에게는 밝혀지지 않는다 하더라도 상인이 이를 복호화해서 결재를 받으므로 상인에 의해 악용될 소지가 있다. 또한, 암호화에 따르는 오버헤더로 인해 작은 가격의 물품에 대해서는 실용성이 없다. 전체적인 비용이나 속도 측면에서는 이후에 기술될 제 3자 개입 방식에 비해 효율적이며, 공개키 방식 암호시스템을 이용함으로써, 고객 인증기능도 추가할 수 있다. 마지막으로 제 3자 개입 방식은 물품 구입시 PIN cord 또는 암호화된 정보를 상인 대신 전자지불을 수행하는 신뢰있는 제3자에게 보냄으로써 인증을 받는다. 고객은 고객카드 번호와 연결되는 PIN을 신뢰있는 제 3자로부터 얻는다. 물품구입시 고객은 제 3자에게 PIN을 보냄으로써 인증을 요구한다. 제 3자는 고객에게 전자우편등을 통해 구매의사를 확인하고 난 후 상인에게 고객의 PIN을 인증한다. 이 방식은 지불 처리 비용이 상대적으로 비싸기 때문에 제 3자는 작은 트랜잭션들을 모아서 한꺼번에 처리함으로써, 작은 트랜잭션들이 자주 일어나는 것을 방지하기도 한다.

2.2 전자현금을 이용한 전자지불 시스템

전자현금은 실제 세계에서의 현금의 기능을 수행하는 전자적인 결재 방식을 의미한다. 전자현금이 화폐로서의 기능을 수행하기 위해서는 여러가지 만족시켜야할 특성들이 있는데, 그것은 실제 화폐나 그에 상응하는 물품으로 바뀔 수 있어야 한다는 것, 교환되는 동안에 쉽게 복사되거나 두번 이상 사용될 수 없어야 한다는 것등이다. 이러한 기본적인 기능들을 포함하여, 전자현금이 만족시켜야 할 여섯가지 특성을 T.Okamoto와 K.Ohta가 [3]에서 나열하였다.

- 전자현금의 안전성은 물리적인 장치에 의존해서는 안된다.
- 전자현금은 복사되거나 재사용될 수 없어야 한다.
- 어느누구도 사용자와 구매를 연결시킬 수 없어야 한다.
- 사용자와 상인간의 프로토콜은 오프라인으로 수행될 수 있어야 한다(즉, 중간에 은행과 같은 제 3자의 개입없이 진행될 수 있어야 한다).
- 전자현금은 사용자들 간에도 전달될 수 있어야 한다.
- 일정액의 전자현금은 더 작은 액수의 전자현금으로 나뉠 수 있어야 한다.

2.3 전자수표 기반의 전자지불 시스템

전자수표는 실세계에서의 수표와 유사하게 동작한다. 신용카드 지불방법에서 제 3자 개입 방식과 같이, 전자수표 기반의 전자지불 시스템은 고객이 서명한 전자수표의 정보들을 검증해 줄 제 3자를 필요로 한다. 전자수표에 서명하기 위해 필요한 정보는 고객자의 이름, 수표, 그리고 고객의 전자서명 등이다. 이러한 정보들은 암호화되어 상인에게 전달되고 상인은 이를 제 3자에게 전송한 후 검증받는다. 제 3자는 수표의 전자서명이 검증되는대로 수표를 상인의 구좌에 적립한다.

이상에서 언급된 지불 시스템 이외에도 소액지불을 위한 전자지불 시스템이나 스마트 카드를 이용한 지불 시스템등이 있다. 특히 소액지불 시스템의 경우, 대금 결제를 위한 트랜잭션당 비용이 거래 금액을 초과할 경우, 매우 비효율적이 되므로 특별한 프로토콜을 필요로 하기도 한다 [4, 5].

3 전자현금 시스템의 특징들

본 절에서는 전자현금에 관련된 주요 특징들에 대해서 설명한다. 전자현금 프로토콜을 특징 지우는 단어들은 다음과 같은 것들이 있다.

- 이중 사용(double spending) 방지
- 추적 불가능성(untraceability)
- 이동가능성(transferability)
- 온라인/오프라인(online/offline)

각각의 문제에 대해서는 수많은 연구들이 진행되어 왔으며, 본 논문에서 구현된 EOEC 시스템을 기술하기 위해 각각에 대해 간략히 설명한다.

3.1 이중 사용 방지

전자 현금과 현행 화폐 제도중의 하나인 지폐 사이에 일어날 수 있는 부정 행위를 살펴보면, 지폐에 있어서는 위조가 있을 수 있으며 전자 현금에 있어서는 이중 사용이 있을 수 있다. 위조는 은행 또는 정당한 발행 기관의 허가없이 돈을 만들거나, 기존의 돈으로부터 새로운 돈을 만들어 내는 행위를 말하며 이중 사용은 전자현금이 전자 정보로 이루어져 쉽게 복사가 가능한 점을 이용하여 1회 사용 후 다시 다른 곳에 동일한 전자 현금을 사용하는 것을 말한다. 전자 현금에 대한 이중 사용 방지는 사용된 전자 현금의 정보로부터 컴퓨터가 동일한 전자 현금을 조사하여 이중 사용자의 계좌번호와 사용자의 신분을 알아내는 방식을 주로 취하고 있다. 온라인의 경우는 이중 사용 방지가 용이하며 즉시 거래를 중지시킬 수 있으나, 오프라인의 경우는 전자 현금 사용전 거래 중지가 곤란하며 추후 부정 사용자 명단에 공개하거나 신용 거래를 중지하는 방안을 취하고 있다.

3.2 추적 불가능성

전자 현금에 있어서 안전성이나 효율성 이외에 가장 중요한 관심사가 되고 있는 것이 개인의 사생활 보호(privacy)이다. 전자현금에 있어서 개인 사생활 보호는 어떤 목적으로 전자 현금을 사용하였거나, 어디서 전자현금을 가져

왔는가에 대한 개인의 비밀 정보를 다른 사람이 알 수 없게 하는 것이다. 전자 현금에서 고려될 수 있는 사생활 보호는 크게 두 종류인데 지불자 이외의 다른 모든 사람들이 결탁한다 하더라도, 지불자가 구매한 정보에 대해서 알 수 없는 **지불자 추적 불가능성(payer untraceability)**과 수취인이외의 다른 모든 사람들이 결탁한다 하더라도, 수취인이 어디로부터 받은 전자 현금인지에 대해서 알 수 없는 **수취인 추적 불가능성(payee untraceability)**이 있다[1].

3.3 이동 가능성

이동 불가능(non-transferable)한 전자현금은 1회 사용 후 곧바로 상점과 은행간의 결재 단계가 이루어져야 하는 것을 말하며, 이에 반해 이동 가능(transferable)한 전자 현금은 발행 단계 이후의 지불 단계가 여러번 이루어지는 즉, 지불자와 지불자간이나 지불자와 상점간의 단계가 복수회 이루어진 후에 결재 단계로 향하는 것을 말한다. 즉, 상점에서 수령한 전자 현금을 바로 은행에 제출하지 않고, 상점 자신이 지불인이 되어 또 다른 상점에 지불될 수 있는 것을 의미한다. 현재 통용되고 있는 화폐 제도는 이동 가능하므로 전자현금도 이동 가능한 것이 바람직하나, 이를 구현할 경우 전자 현금을 위한 정보량이 이동되는 횟수에 비례하여 증가하게 되는 단점이 있다.

3.4 온라인/오프라인

온라인은 고객 관리 및 전자현금 관련 정보를 수록한 거대한 데이터베이스를 유지하여, 매 지불 단계시마다 허가를 해주는 중앙 허가 기관 즉 은행과 모든 참가자가 접촉하는 것을 말한다. 다시 말해서 지불 단계와 결재 단계가 동시에 행하여지는 것을 말하며, 이중 사용을 지불 단계에서 사전에 방지할 수 있는 장점이 있으나, 많은 통신량이 한곳으로 집중화되는 문제점과 거래에 따른 통신 비용이 증가하게 되는 문제점이 생기게된다. 이에 반해, 오프라인은 지불 단계와 결재 단계가 동시에 이루어지지 않은 형태이며 일정 시간 경과후 수신된 전자 현금을 일괄 처리하여 은행에 결재 요구하는 것으로, 이중 사용이 이루어지고 난 후에야 은행에서 이중 사용자에 대한 신분 검출이 가능한 점이 문제점으로 생각할 수 있다. 그러나 통신량의 집중화 방지와 거래에 따른 통신 비용은 적게 소요된다. 두 방식을 응용면에서 고려해보면, 온라인은 고액거래로 높은 안전성을 요구하면서 운용비에 대한 부담이 크게 중요하지 않은 현금 시장에 적합하다. 오프라인은 많은 양의 소액거래가 이루어지는 곳으로 이중 사용으로 인한 부정 가능 금액이 소규모이고 운용비 부담이 문제가 되는 곳에 적합하다.

4 EOEC 시스템 설계 및 구현

본 논문에서는 기존의 지불 방식인 신용카드와 현금을 대신할 수 있는 온라인 전자현금 시스템을 설계하고 구현하였다. 전자현금 시스템의 구성요소로는 고객, 상인, 은행 등이 있으며, 필요하다면 중간브로커나 인증기관을 도입할 수 있다. 본 논문에서 구현한 전자현금 시스템에는 기존의 암호프로토콜 중에서 적합한 프로토콜을 선택 구현하였으며 필요한 경우 프로토콜을 개선하였다.

[1]지불자 추적 불가능성은 자신의 계좌번호와 연결되어 있는 발행 단계와 지불 단계가 서로 연결될 수 없음을 의미하여, 수취인 추적 불가능성은 결재 단계와 지불 단계가 연결될 수 없음을 의미한다

전자현금에서 가장 문제가 되고 있는 것들은 이중사용(double spending)과 잔돈처리 문제이다. 이중사용을 방지하기 위한 가장 간단한 방법은 고객이 상점에서 상품을 구입할 때, 상점에서 결재하기 전에 그 전자현금이 사용된 적이 있는지를 확인하는 방법이다. 그러나, 이 방법은 여러가지 문제점을 갖고 있다. 우선, 은행에서는 방대한 데이타베이스를 관리해야 하는 부담이 생긴다. 그리고, 각 고객들은 상품구입시에 상점과 은행간의 통신 및 은행에서의 데이타베이스 검색에서 발생하는 시간지연을 감수해야 하는 불편이 따른다. 또한, 모든 지불 트랜잭션마다 상점과 은행간의 통신이 필요하므로, 은행은 Hot spot이 되어 병목현상이 발생할 수 있다.

이러한 문제점들을 감소시키기 위해 다음과 같은 방법이 이용되기도 한다. 고객이 대금을 결재할 때, 상점이 임의의 이진수열을 제시하면 고객은 이에 맞는 대답을 해야하고, 이때 주고받은 정보들이 은행에 저장된다. 만약 고객이 같은 전자현금을 두 번 이상 사용하게 되면, 고객과 상인이 주고받은 정보들에 의해 돈의 주인이 밝혀지게 된다. 이는 방대한 데이타베이스의 필요성을 없애 주는 것은 아니지만, 오프라인으로 일괄처리할 수 있으므로 통신 선로상의 병목현상을 막을 수 있고, 통신시간과 데이타베이스 검색으로 인한 시간지연이 없어진다. 하지만, 이는 이중사용을 나중에 적발할 수 있도록 하는 것이지, 이중사용을 막을 수는 없다. 따라서, 이 방법을 사용하기 위해서는 제도적 장치가 뒷받침 되어야 하므로 본 논문에서는 온라인으로 이중사용을 막는 방법을 택하였다.

앞에서 지적한 바와 같이 온라인으로 데이타 베이스를 검색함으로써 이중사용을 방지하는 방법은 병목현상과 시간지연의 문제점이 따르는데, 이를 해결하기 위해서 본 논문에서는 분산시스템을 설계한다. 신용카드 기반의 전자지불 시스템에서 PIN으로 제 3자를 할당했듯이, 전자현금 시스템에서는 전자현금을 나타내는 이진수열로 bank agent를 할당하도록 한다. 그림 1에 본 논문에서 가정하는 시스템을 나타내었다.

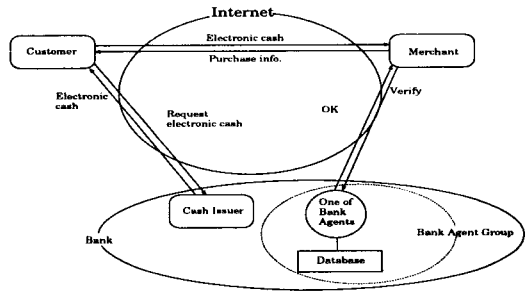

그림 1: 이중사용 방지를 효율적으로 수행하기 위한 전자현금 분산구조

그림 1과 같은 시스템에서 고객은 은행에서부터 전자현금을 인출한다. 고객이 상품을 구입하고자 할 때, 상인에게 자신이 인출한 전자현금을 제시한다. 상인은 이를 bank agent에게 forwarding하여 이중사용 여부를 조사한 후 이상이 없으면 고객에게 구매정보를 돌려줌으로써 지불절차를 완료한다.

4.1 프로토콜 구성 요소

본 절에서는 구현한 온라인 전자현금 프로토콜의 전체적인 동작 과정과 이를 구성하는 세가지 개체인 은행(bank)과 고객(customer) 그리고 상인(merchant)의 구현 내용을

세히 설명한다. EOEC 시스템은 전체적으로 다음과 같이 다섯가지 개체로 구성된다.

- Cash Issuer
- Cash Clearer
- Certificate Authority
- User(Customer)
- Merchant

이러한 다섯가지 개체들의 동작상황을 그림 2에 나타내었다. 먼저 Cash Issuer는 User에게 전자 현금을 발행하는 역할을 수행한다. User는 Cash Issuser에게로부터 발행받은 전자 현금을 이용하여, Merchant에게서 자신이 사용하는 전자 현금이 올바른 과정을 거쳐 발행되었고, 이중사용되지 않았는지를 확인받은 후에 대금을 지불한 후 상품을 구매한다. Merchant는 User가 제시한 전자 현금이 이중 사용되지 않았는지를 Cash Clearer를 통해서 확인받는다. Cash Clearer는 Merchant로부터의 요구를 받아서 확인해주고, 확인된 돈을 실제 돈(real money)로 바꾼 후, Merchant의 계좌에 넣어주는등의 일을 수행한다. 이러한 동작 과정에서 모든 개체들은 각각의 공개키(public-key)를 필요로하게 되는데 이때 필요한 키는 인증기관(Certificate Authority)으로부터 건네 받는다. 인증기관는 자신의 각 개체로부터 요구되는 공개키를 자신의 개인키(private-key)를 이용하여 서명함으로써 각 개체들이 받게 되는 키가 올바른 인증기관으로부터 발행된 것임을 믿게 한다. 이에 대해 각 개체들은 수취한 키를 인증기관의 공개키를 이용하여 복호화해 봄으로써 진위 여부를 확인할 수 있다. 공개키 확인시에 동작하는 이와 같은 일련의 과정을 전자서명(digital signature) 기법이라고 한다. 그리고 각 개체에 부착되어 있는 공개키 에이전트(public-key agent)가 이러한 역할을 수행하게 되는 것이다. 편의상 그림 2에는 Cash Issuer에만 공개키 에이전트를 표시해 두었다.

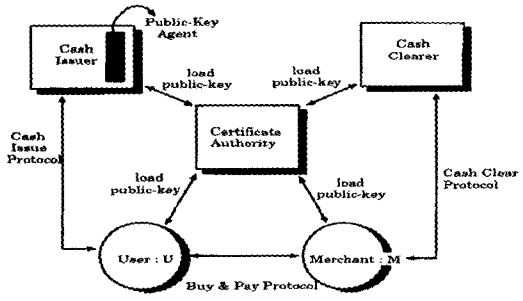

그림 2: EOEC 시스템

각 개체들간의 상호 동작들은 세가지 암호학적 알고리즘과 기법들이 사용되는데 그것들은 RSA 암호알고리즘, 맹목서명(blind signature), 그리고 cut-and-choose 기법이다. 이들 알고리즘과 기법에 대해서 각각 살펴본다.

4.1.1 RSA 암호알고리즘

RSA 공개키 암호 시스템은 약 200자리 정수의 소인수 분해의 어려움에 그 안전성을 근거하고 있다 [6, 7]. 이 RSA 암호 시스템에서는 오일러의 정리가 쓰이는데 이를 간략하게 살펴보자.

양의 정수의 집합 $\{1, 2, 3 \cdots, n-1\}$의 원소들 중에서 n과 서로소의 관계에 있는 원소들의 개수를 $\varphi(n)$으로 나타내고, 이를 오일러의 φ-함수라고 한다. 특별히 소인수 p에 대해서 $\varphi(p) = p - 1$이다. 큰 정수 n에 대하여 $\varphi(n)$값을 구하기 위해서는 n의 소인수 분해가 필수적이다. 즉, n이 두 소수 p와 q의 곱일 때 $\varphi(n) = (p-1)(q-1)$이다. 따라서 소인수분해 없이 $\varphi(n)$의 값을 구하기는 매우 어렵다. 오일러의 정리란 서로소인 두 양의 정수 a와 n에 대하여,

$$a^{\varphi(n)} \equiv 1 (mod\ n) \quad (1)$$

이 성립한다는 것이다. $\varphi(n)$ 자체는 오일러의 totient 함수라고 한다.

다음에 RSA를 이용한 암호화, 복호화 단계를 간략히 서술한다.

(1) 두개의 큰 소수 p와 q를 선정하여 자신의 비밀키로 한다.

(2) $n = pq$인 n을 공개하고, $\varphi(n)$과 서로소인 임의의 정수 e를 선택하여 공개키로 한다.

(3) $e \times d \equiv 1 (mod\ \varphi(n))$이 되는 d를 유클리드 호제법등으로 계산하여 비밀키로 한다. 즉, p와 q, 그리고 d는 비밀키가, n과 e는 공개키가 되는 것이다.

< 암호화 단계 >

평문 M을 공개키 e를 사용하여 M^e을 계산한 다음 modulus n으로 나누고 이때 암호문 C는 다음과 같다.

$$C \equiv M^e (mod\ n)$$

< 복호화 단계 >

암호문 C를 비밀키 d를 사용하여 C^d한 다음 modulus n으로 간단히 한다. 복호화를 거쳐 평문이 나오게 되는 관계식은 다음과 같다.

$$C^d \equiv (N^e)^d = M^{t\varphi(n)+1} = M^{t\varphi(n)} \times M \equiv M (mod\ n)$$

여기서 t는 $e \times d \equiv 1 (mod\ \varphi(n))$에서 유도되는 $ed = t\varphi(n) + 1$을 만족하는 정수이다.

4.1.2 맹목서명(Blind Signature)

맹목서명은 전자서명의 익명성을 제공하는데 사용되는 암호학적 기술이다. 맹목서명의 개념은 David Chaum에 의해서 처음 언급되었다 [8]. 이는 RSA 알고리즘을 이용한다. 아래의 프로토콜로써 맹목서명의 개념을 설명하고자 한다.

A는 공개키 e와 비밀키 d, 그리고 공개 modulus n을 가지고 있다. 그리고 B는 메시지 m에 대해서 맹목적으로 사인(blindly sign)하고자 한다.

(1) B는 1과 n사이의 임의의 랜덤 값 k를 선택한다. 그리고 다음의 값을 계산한다.

$$t = mk^e \ mod\ n$$

(2) A는 다음의 식으로 t를 사인한다.

$$d^d = (mk^e)^d \ mod\ n$$

(3) B는 다음의 식을 계산함으로써 t^d을 unblind시킨다.

$$s = t^d / k \ mod\ n$$

(4) 결과로 나온 식은 다음과 같다.

$$s = m^d \ mod\ n$$

(4) 단계에서의 결과는 쉽게 증명될 수 있다.

$$t^d \equiv (mk^e)^d \equiv m^d k (mod\ n) \quad (2)$$

$$so, t^d/k = m^d k/k \equiv m^d (mod\ n) \quad (3)$$

Chaum은 좀 더 복잡한 일련의 맹목 서명 알고리즘들을 그 후 제안하였는데, 이것들은 "blind unanticaipated signature"라고 불리운다. 이러한 알고리즘들은 좀 더 복잡한 반면, 더 많은 융통성을 가진다.

4.1.3 Cut-and-Choose

맹목서명과 함께 전자서명의 익명성을 위해 사용되는 기법으로, n개의 문서를 받으면, 그 중 $n-1$개를 확인하고 나머지 한 개는 확인하지 않은 상태에서 맹목서명을 수행하는 기법을 의미한다. 이 때, 확인하는 $n-1$개는 임의로 선택한다. 안전성의 증가를 위해 $\frac{n}{2}$개만 확인하고 나머지 $\frac{n}{2}$개를 묶어서 서명할 수도 있다.

4.2 변수 정의 및 자료 구조

본 절에서는 실제 EOEC 시스템을 구현하는데 있어서 사용된 변수와 자료 구조를 설명한다.

변수 정의

1. e_B : Bank의 공개키(public-key)
2. N_B : Bank의 common modulus
3. d_B : Bank의 개인키(private-key)
4. M : money, "struct_money" 형식
5. $M(Amount)$: money "M"의 "amount" field
6. e_M : Merchant M의 공개키
7. N_M : Merchant M의 common modulus
8. d_M : Merchant M의 개인키
9. $M(bank\_name)$: money "M"의 "bank_name" field
10. $M(hash)$: money "M"의 "sequence number" field

자료 구조

1. typedef struct _money {
   Amount amount;
   char bank_name[MaxSzId];
   INTG seq_num;
   INTG hash;
   INTG signature;
   }; Money
2. typedef struct _INTG {
   int leng;
   DIGIT cont[MaxLengIntg];
   } INTG;
3. typedef unsigned short DIGIT;
4. typedef long Amount;

4.3 프로토콜

그림 2에서 볼 수 있듯이 EOEC 시스템은 전체적으로 다음과 같은 세가지 프로토콜에 의해서 동작한다.

- Cash Issue Protocol
  은행의 cash issuer 모듈과 사용자 간의 현금 발행 프로토콜로서, 사용자가 은행으로부터 전자현금을 발행받고, 은행은 발행한 만큼의 액수를 사용자의 계좌에서 감하는 동작을 수행한다.
- Buy & Pay Protocol
  사용자와 상인(merchant)간의 물건 구매 프로토콜로서, 사용자는 상인에게 전자현금을 지불하고, 상인으로부터 상품이나 서비스를 받는 프로토콜이다.
- Cash Clear Protocol
  상인과 은행간의 현금 정산 프로토콜로서, 상인은 고객으로부터 받은 전자현금을 은행에게 제공하고, 은행은 상인이 제시한 전자현금의 해당 액수만큼을 상인의 계좌에 입금한다.

Cash clear protocol을 수행할 때, 은행의 현금 정산 서버가 단일 서버라면 이는 병목현상을 유발한다. 따라서, 본 논문에서는 현금 정산 서버를 두 단계로 분리하여, 한 단계는 상인으로부터 온 정산요구를 해당하는 정산서버로 진행(forwarding)하는 역할을 하고, 정산버서는 이중사용을 조사한 후 정산하는 작업을 수행한다. 각 프로토콜들을 자세히 살펴본다.

4.3.1 Cash Issue Protocol

Cash issue protocol은 은행의 cash issuer 모듈과 사용자 간의 현금 발행 프로토콜로서, 사용자가 은행으로부터 전자현금을 발행받고, 은행은 발행한 만큼의 액수를 사용자의 계좌에서 감하는 동작을 수행한다. Cash issue protocol은 RSA를 이용한 blind signature와 cut-and-choose 기법을 사용하여 구현되었다. 다음은 프로토콜의 동작 과정을 설명한다.

User: U　　Bank(Cash Issuer)

$R_i^{e_B} M_i \ mod\ N_B = C_i$

prepare $< C_1, C_2, \cdots, C_{100} > = \vec{C}$

$ID_U, \vec{c}$, **amount**

$\longrightarrow$

select random number r
(where $1 \leq r \leq 100$**)**

r

$\longleftarrow$

assume $\vec{R} = < R_1, R_2, \cdots, R_{r-1}, R_{r+1}, R_{100} >$

$\vec{R}$

$\longrightarrow$

compute $(C_i^{d_B} \times R_i^{-1})^{e_B}\ mod\ N_B \equiv M_i$
check whether $M_i(Amount) = amount$**?**
if correct, compute $C_r^{d_B}\ mod\ N_B$

$C_r^{d_B}\ mod\ N_B$

$\longleftarrow$

$C_r^{d_B}\ mod\ N_B \equiv R_r \times M_r^{d_B}\ mod\ N_B$
compute $(C_r^{d_B}\ mod\ N_B) \times R_r^{-1}\ mod\ N_B$
save $M_r^{d_B}\ mod\ N_B$

withdraw "amount" from ID_U**'s account**

4.3.2 Buy & Clear Protocol

이 프로토콜은 User가 Cash Issuer로부터 발행받은 전자화폐를 이용하여 merchant로부터 물건을 구매할때에 동작하는 프로토콜이다. 다음의 그림은 Buy & Clear 프로토콜을 나타내고 있다.

User: U　　Merchant: M

---

load $M_\tau^{d_B}$ *mod* N_B
compute $(M_\tau^{d_B}$ *mod* $N_B)^{e_M}$ *mod* N_M

$(M_\tau^{d_B}$ *mod* $N_B)^{e_M}$ *mod* N_M

$\longrightarrow$

compute
$(M_\tau^{d_B}$ *mod* $N_B)^{e_M \times d_M}$ *mod* $N_M) \equiv M_\tau^{d_b}$ *mod* N_B
check whether
$(M_\tau^{d_B})^{e_B}$ *mod* $N_B(bank\_name) \equiv M_\tau(bank\_name)$
check Double Spending by using "Cash Clear Protocol"

Goods

$\longleftarrow$

remove $M_\tau^{d_B}$ *mod* N_B

4.3.3 Cash Clear Protocol

Merchant M은 User U의 전자화폐가 중복 사용되지 않았는가를 확인해야 하는데, 이때 Merchant M과 Bank의 Cash Clearer사이에서 이 프로토콜이 동작한다.

Merchant: M　　Cash Clearer

---

compute $M_\tau^{e_B}$ *mod* N_B

$M_\tau^{e_B}$ *mod* N_B

$\longrightarrow$

compute $(M_\tau^{e_B})^{d_B}$ *mod* $N_B \equiv M_\tau$
check $M_\tau(hash)$ **is in Used Money Set(UMS)**
if not

"OK"

$\longleftarrow$

add $M_\tau(hash)$ **to Used Memory Set(UMS)**

4.4 Bank Agent Group

제 2절에서 여러가지 지불 프로토콜들을 살펴보면서 이중 사용 방지를 위해서 온라인으로 각 거래 트랜잭션들을 처리하고자 하는 목표를 세웠다. 하지만 온라인으로 처리할 경우의 문제점은 cash clear protocol을 수행할 때, 은행의 현금 정산 서버가 단일 서버라면 이는 병목현상을 유발한다. 따라서, 본 논문에서는 현금 정산 서버를 두 단계로 분리하여, 한 단계는 상인으로부터 온 정산요구를 해당하는 정산서버로 진행(forwarding)하는 역할을 하고, 정산버서는 이중사용을 조사한 후 정산하는 작업을 수행한다. 본 논문에서 구현한 2단계 은행 구조는 그림 <그림 3>과 같다.

우선 각 은행에는 Cash Clearing Agent가 설치되어 있다. 이제 상인이 전자 현금을 clearing하는 과정에서 가장 가까운 은행으로 트랜잭션을 보내면 그 은행의 Cash Clearing Agent는 그 거래에서 사용된 전자 현금의 sequence number field를 보고 어느 은행인지를 판단한 후에 해당 은행으로 거래의 처리를 넘기게 된다. 이렇게 동작하기 위해서는 각 은행마다 발행하는 전자 현금의 sequence number의 범위가 미리 약속으로 정해져 있어야 한다.

5 결론과 향후 연구과제

본 논문에서는 온라인 전자 현금 시스템을 설계하고 구현하였다. 이를 위해서 우선 기존의 여러가지 전자 지불 시스템에 대해서 살펴 보았고, 특히 전자현금 시스템에서의 이중 사용 방지와 온라인 트랜잭션 처리에 관한 기존 시스템의 문제점을 살펴 보았다. 전자현금 시스템에서 이중 사용 방지를 위한 온라인 트랜잭션 처리의 가장 큰 문제점은 중앙 은행으로 각 거래를 위한 통신량이 집중하여, network bottleneck이 생길 수 있다는 점이다.

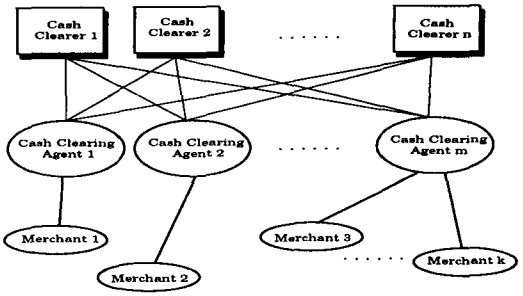

그림 3: "Clearing" 연산을 위한 2중 은행 구조

본 논문에서는 이를 해결하기 위하여, 2중 은행 구조를 제안하고 구현하였다. 이 구조의 가장 큰 특징은 각 은행마다 그 은행에서 발행하는 전자현금을 구분하기 위해서 전자 현금의 정보에 들어 있는 일련번호(sequence number)의 값을 은행별로 범주화한다는 점과, 이를 지원하기 위해서 각 은행마다 Cash Clearing Agent라는 모듈을 가진다는 점이다. 이와 같은 방법을 사용함으로써, 거래때 주고 받는 전자 현금은 발행한 은행에서만 처리하면 되므로 중앙 은행으로의 통신량 집중 현상은 막을 수 있다.

현재는 Unix를 운영체제로 사용하는 웍스테이션상에서 동작할 수 있도록 구현되었으나, 실제로 사용되기 위해서는 다양한 플랫폼에서 돌아갈 수 있어야 한다. 또한 웹과의 연동등 인터페이스 개발 또한 향후 연구과제이다.

참고 문헌

[1] First Virtual. *http//:www.fv.com.* Internet document, 1996.

[2] DigiCash. *http//:www.digicash.com.* Internet document, 1996.

[3] T.Okamoto and K.Ohta. Universal electronic cash. In *CRYPTO'91*, pages 324–347, 1992.

[4] S.Glassman et al. *The millicent protocol for inexpensive electronic commerce.* Digital Equipment Corp., 1995.

[5] A.Furche and G.Wrightson. *SubScrip - An efficient protocol for pay-per-view paments on the internet.* The University of Newcastle, 1996.

[6] R.L.Rivest, A.Shamir, and L.Adleman. A method for obtaining digital signatures and public key crytosystems. *CACM*, 21:120–126, 1978.

[7] A.Selby and C.Mitcheil. Algorithms for software implementations of RSA. *IEE Proceedings(E)*, 136(3):166–170, MAY 1989.

[8] D.Chaum. Blind signatures for untraceable payments. In *CRYPTO'82*, pages 199–203, 1993.