DIT(Digital Investment Trust) 시스템 설계를 위한 계좌관리

정은희*, 이병관**
*관동대학교 전자계산공학과
**관동대학교 컴퓨터공학과
e-mail:jeongnala@hanmail.net
bklee@mail.kwandong.ac.kr

Account Management for the Design of DIT(Digital Investment Trust) system

Eun-Hee Jeong*, Byung-Kwan Lee**
*Dept of Computer Science, Kwandong University
**Dept of Computer Engineering, Kwandong University

요약

본 논문에서 제안하는 DIT(Digital Investment Trust) 시스템은 전자상거래 결재 방법으로 이용되는 디지털 캐쉬 시스템을 확장한 개념으로서, 소액지불뿐만 아니라, 계좌 생성 및 계좌 이체를 할 수 있으며, 투자 신탁의 개념을 도입한 인터넷 기반의 은행업무 프로젝트이다. DIT 시스템에선 계좌 생성과 계좌 이체에 암호화 알고리즘인 ADES를 이용하여 고객의 정보를 암호화시켰으며, RSA 알고리즘을 이용하여 전자서명을 하여 고객의 정보와 자산을 제 3자로부터 보호하였다.

1. 서론

현재 상용화되고 있는 디지털 캐쉬 시스템은 현금을 전자화 하는 개념으로서, 전자데이터에 현금과 같은 가치를 부여하는 방법으로 전자 상거래 결제 시스템에 주로 사용되고 있다.

디지털 캐쉬 시스템이 전자 상거래 결제방법에 이용되는 것은 현금 또는 신용카드에 의한 결제가 갖는 문제점인 소액지불, 개인정보 누출 등을 해결할 수 있기 때문이다. 디지털 캐쉬 시스템이 갖는 특징들은 첫째, 결제 비용의 절감이고 둘째, 안정성 향상, 지속성을 확보할 수 있으며, 셋째, 기존의 금융제도와의 호환성을 들 수 있다[1][2].

본 논문에서 제안하는 DIT(Digital Investment Trust) 시스템은 이러한 디지털 캐쉬 시스템을 확장한 개념으로서, 소액지불뿐만 아니라, 계좌 생성 및 계좌 이체를 할 수 있으며, 투자 개념을 도입한 인터넷 기반 은행업무 프로젝트이다.

본 논문에서 제안한 DIT 시스템을 크게 세 가지 부분으로 나누어 볼 수 있는데, 첫째 DIT 시스템 내에서 계좌 생성과 거래를 익명화 하였고, 둘째 고객이 인터넷을 통해 안전하게 접근하여 화폐를 전송하는 것을 허용하였으며, 셋째 고객이 예금한 자금으로 투자를 하여 이윤을 남기는 투자 신탁이다.

본 논문에선 DIT 시스템의 계좌 관리에 대해서만 설명을 하며, 논문의 구성은 제 2장에서 DIT 시스템의 설계에 대해서 설명하였고, 제 3장에선 DIT 계좌 생성과 계좌이체를 설명하였다. 마지막으로 제 4장에선 결론 및 향후 연구 방향을 제시하였다.

2. DIT 시스템 설계

2.1 ADES(Advanced Data Encryption Standard)

ADES는 기존의 암호화 알고리즘인 DES를 변형시킨 것으로, 암호화 하고자하는 메시지와 암호화 시키는 키의 값을 각각 결합하는 과정에서 기존의

DES는 하나의 고정된 IP(Initial Permutation) 박스를 기본 16라운드 순환할 때마다 같은 IP 박스를 사용하였는데, ADES는 각 라운드마다 IP 박스를 각각 다르게 설정함으로서 도청자에 의해 하나의 IP 박스를 도청당하더라도 전체의 암호문 혹은 복호문을 알 수 없기 때문에 보다 강력한 암호화 알고리즘이다.

2.2 RSA(Rivest, Shamir, Adleman)

RSA 공개키 암호화 방식은 메시지 암호화 및 전자서명을 위한 알고리즘이다. RSA는 공개키와 비밀키를 한 쌍으로 사용하는데 공개키는 공개되어 많은 사람들이 알 수 있는 경우이고, 비밀키는 본인만이 알 수 있도록 비밀을 유지한다. 따라서 이 암호 방식은 공개키만 사용되고 비밀키는 전송되지 않기 때문에 데이터가 전송 도중에 가로채기를 당하거나 변조될 염려가 없다.

RSA가 전자서명에 이용될 때에는 해시 함수를 사용하기 때문에 메시지의 무결성(integrity)까지도 보장해 줄 수 있다.

2.3 사용자 모듈

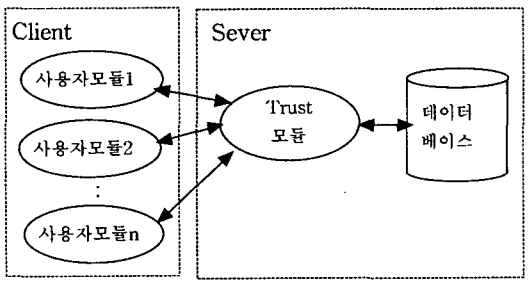
사용자 모듈은 Trust 서비스에 접근하려는 일반적인 웹 브라우저와 관련된 작업을 하며, 고객번호를 생성하고 저장한다.

2.4 Trust 모듈

Trust 모듈은 DIT 시스템을 제어하기 위해 웹서버와 관련된 작업을 하며, Trust 번호를 생성하고 기록을 작성한다. 또한 익명 계좌에 대해 예금 혹은 인출을 허용한다.

2.5 데이터베이스

암호화 알고리즘인 ADES를 이용해 Trust 계좌 정보를 저장하며, ADES에 이용된 대칭 암호키 s를 RSA기법을 이용해 암호화하여 저장한다.



[그림 1] DMT 시스템의 구성도

3. DIT 시스템 시뮬레이션

3.1 DIT 계좌 생성

DIT 계좌를 개설하는 것은 계좌에 현금을 입금시키는 것만큼 간단하다. 고객은 익명이므로 DIT는 고객 정보가 필요하지 않으며, 계좌 번호는 고객의 사용자 모듈에 의해 생성되어, Trust 모듈에 의해 예금 기록과 함께 DIT 데이터베이스에 저장된다.

[그림 2]는 사용자 모듈이 Trust 모듈에 계좌 생성을 요청하는 알고리즘이다. 계좌 번호는 사용자 모듈에 의해 생성되며, 계좌번호와 금액에 대한 정보는 DIT의 공개키로 암호화되어 전송된다.

```
User_module(){
  generate_private_key(x);
  y = g^x mod p;
  y1 = SHA_1(y);
  send_to_trust(c_flag, y1, C);
  call_deposit(y);
  send_to_DIT(c_flag, C, y1, y);
  if(SHA_1(y)== s1^e mod n)
     save_user(y, C, s);
}
```

[그림 2] User Module 알고리즘

[그림 3]은 고객이 요청한 계좌 개설을 승인하는 알고리즘으로 전자서명에 RSA 서명 알고리즘을 사용하였다.

```
Trust_module(){
  save_to_database(c_flag, y1, C);
  while(!EOF){
    if(SHA_1(y) == y1){
       generate_RSA_key(d, e);
       y1 = SHA_1(y);
       s1 = y1^d mod n;
       send_to_user(y, s1);
    }
  }
}
```

[그림 3] Trust Module 알고리즘

3.2 DIT 데이터베이스

[그림 4]는 계좌 개설이 승인된 정보를 DIT 데이터베이스에 저장하는 알고리즘으로 고객 정보의 보안을 위해 ADES 암호화 기법을 사용하였으며, 대칭키를 RSA로 한번 더 암호화하여 보안을 강화시켰다.

```
DIT_database(){
  s = generate_ADES_key();
  nonce = generate_nonce();
  c = concatenate(C, y, nonce);
  v = ADES(c);
  s1 = RSA(s);
  save_to_DIT_db(i, v, s1);
}
```

[그림 4] DIT 데이터베이스 알고리즘

3.3 DIT 계좌 이체

DIT 시스템의 고객들끼리 계좌 이체를 하는 경우로, 누구든지 paying 고객 혹은 receiving 고객이 될 수 있다.

[그림 5]는 paying 고객에 대한 알고리즘으로 Trust 모듈에 계좌이체에 대한 정보를 전송하여 계좌이체를 한 후, Trust 모듈로부터 계좌 잔액에 대한 정보를 전송 받아 paying 고객의 컴퓨터에 계좌번호와 금액, DIT의 서명을 저장한다.

```
pay_module(){
  t = concatenate(y1, y, C, T);
  send_to_DIT(t);
  generate_RSA_key(d, e);
  s = concatenate(C, y);
  s1 = SHA_1(s);
  s2 = s1^d mod n;
  send_to_DIT(s, s2);
  s3 = SHA_1(c);
  s4 = s3^d mod n;
  send_to_DIT(c, s4);
  if(SHA_1(c1) == s5^e mod n){
    save_to_user(y, C-T, s5);
  }
}
```

[그림 5] paying 고객 알고리즘

[그림 6]은 receiving 고객에 대한 알고리즘으로, receiving 고객의 계좌 정보를 paying 고객에게 전송하고, receiving 고객의 계좌에 화폐를 계좌이체 하는 알고리즘이다.

```
receive_module(){
  generate_private_key(x);
  y* = g^x mod p;
  y1 = SHA_1(y*);
  send_to_pay_user(y1);
  send_to_DIT(T, y1, y*);
  generate_RSA_key(d, e);
  c1 = SHA_1(c)
  s = c1^d mod n
  send_to_DIT(c, s);
  if(SHA_1(t1) == s1^e mod n){
    save_to_user(y*, T, s1);
  }
}
```

[그림 6] receiving 고객 알고리즘

[그림 7]은 paying 고객이 Trust 모듈에 paying 고객의 계좌 정보를 전송하여 DIT 데이터베이스의 정보를 갱신하는 알고리즘이다.

```
Trust_module1(){
  if(SHA_1(s) == s2^e mod n){
    c = generate_number();
    send_to_pay_user(c);
  }
  if(SHA_1(c) == s4^e mod n){
    serach_db(no, D, s');
    replace_db(y, C-T);
    send_to_pay_user(y, C-T);
    generate_RSA_key(d, e);
    c1 = concatenate(C-T, y);
    c2 = SHA_1(c1);
    s5 = c2^d mod n;
    send_to_pay_user(c, s5);
    save_to_DIT_db(c2, y, C-T);
    save_to_database(T, y1);
  }
}
```

[그림 7] Trust Module 알고리즘 1

[그림 8]은 Trust 모듈이 receiving 고객의 정보를 전송 받아 DIT 데이터베이스의 receiving 고객의 정보를 갱신하는 알고리즘이다.

```
Trust_module2(){
  search_db(T, y1);
  if(SHA_1(y*) == y1){
     c = generate_number();
     send_to_receive_user(c);
  }
  if(SHA_1(c) == s^e mod n){
     t1 = concatenate(T, y*);
     s = SHA_1(t1);
     s1 = s^d mod n;
     send_to_receive_user(t1, s1);
     save_to_DIT_db(s, y*, T);
  }
}
```

[그림 8] Trust Module 알고리즘 2

4. 결론

본 논문에서 제시한 DIT 시스템은 단순한 기능의 디지털 캐쉬 시스템이 아니라 익명 계좌 생성 및 계좌 이체와 투자 개념이 도입된 인터넷 기반 은행 업무 시스템이다.

DIT 시스템에선, 고객 모듈과 Trust 모듈 사이에서 데이터를 전송할 때, 공개키 암호화 방식을 사용하였으며, 특히, DIT 데이터베이스에 고객 정보를 저장할 때 비밀키 암호화 방식인 DES 알고리즘의 변형인 ADES를 사용하였고, 대칭키를 다시 한번 RSA 알고리즘으로 암호화시킴으로써 고객 정보 보안을 강화시켰다.

또한, DIT 시스템은 계좌 생성 및 계좌 이체를 익명으로 처리하므로, 고객의 정보와 자산을 보호할 수 있는 차세대 금융 관리 시스템이다.

향후 과제로는 본 논문에서 제안한 DIT 시스템에 자산 관리 기능을 추가시키는 것이다.

[참고문헌]

[1] 이은철, "뉴밀레니엄 시대의 전자화폐(상)", 지식재산21, 통권 제59호, 2000년 3월

[2] 이은철, "뉴밀레니엄 시대의 전자화폐(하)", 지식재산21, 통권 제60호, 2000년 5월

[3] 이병관, "전자상거래 보안", 남두도서

[4] 이만영외 5인 공저, "전자상거래 보안기술", 생능출판사

[5] Dabid Pointchebal and Jacques Stern, "Security Proofs for Signature Schemes" Advances in Cryptology-Proceedings of EUROCRYPT'96, pp 387-398

[6] 배움닷컴, http://www.baeoom.com

[7] "PKCS#1 : RSA Encryption Standard", pp. 10-11