프록시 재암호화 기반의 안전한 전자지불시스템

고웅† · 곽진††

요 약

현재 IT와 금융거래를 위한 기술의 발전으로 인해 인터넷과 같은 개방형 통신망을 이용한 효율적인 전자금융 서비스가 사용되고 있다. 이와 같은 환경에서 이용되는 SET, SSL/TLS 등과 같은 프로토콜은 신용카드 기반의 전자결제를 안전하고 효율적으로 수행하기 위해 개발된 전자지불 프로토콜이다. 그러나 대부분의 사용자들은 이에 대하여 정확히 알지 못하며, 보안상의 문제점을 인지하지 못한다. 특히, 이러한 프로토콜은 제품의 구매 및 결제를 위해 세션키를 추가로 생성하여 이용해야하는 등의 키 관리 문제가 존재한다. 따라서 본 논문에서는 이와 같은 문제점을 해결하기 위하여 프록시 재암호화 기법을 이용하여 추가적인 세션키 생성 없이 지불정보 및 주문정보를 송·수신할 수 있는 전자지불시스템을 제안한다.

주제어 : 프록시 재암호화, 전자지불시스템, 키 관리

Proxy Re-encryption based Secure Electronic Transaction

Woong Go† · Jin Kwak††

ABSTRACT

Presently, Enhanced electronic financial service are offered used open network due to development of IT and financial transactions. The protocol in this environments such as SET, SSL/TLS, and so on are electronic transaction protocol to perform electronic payment securely and efficiently. However, most users still does not know accurately how to use and potential problems. It especially has key management problem about generate session key for purchase products or payment. To solve this problem, we propose proxy re-encryption based secure electronic transaction to transmit payment and order information without addition session key.

Keywords : Proxy Re-encryption, Secure Electronic Transaction, Key Management

† 학생회원: 순천향대학교 정보보호학과 대학원 박사과정
†† 종신회원: 순천향대학교 정보보호학과 교수(교신저자)
논문접수: 2011년 12월 04일, 심사완료: 2012년 01월 03일, 게재확정: 2012년 01월 16일

1. 서론

IT와 네트워크의 발전은 전자금융거래의 폭발적인 발전을 가져왔으며, 도입기, 성장기를 거쳐 현재는 안정화 및 고도화 단계에 들어섰다고 볼 수 있다. 이와 같은 추세로 인해 전자금융거래를 이용한 소비 패턴은 우리 생활의 일부로 자리 잡은 상황이다. 특히 대규모의 사이버 쇼핑몰, 소셜 커머스(Social Commerce) 등의 서비스는 현재 가장 각광받는 분야로 그 규모가 해마다 증가하고 있다. 다양한 사용자가 인터넷을 통한 전자결제를 하면서 안전하게 서비스를 이용하기 위하여 필수적으로 구축되어야 하는 것이 전자지불시스템이다. 특히 VISA와 MASTER 카드사가 안전한 신용카드 기반 전자상거래를 구축하기 위해 제안한 전자지불시스템 표준 프로토콜인 SET(Secure Electronic Transaction)가 꾸준히 사용되고 있다[1].

하지만 대부분의 사용자는 자신이 이용하고 있는 서비스가 어떠한 절차를 거쳐 제공되고, 결제되는지에 대하여 충분한 습득이 어려운 실정이다. 이러한 이유는 대부분 서비스 제공의 측면에서만 정보를 공개하고 있고, 실제 진행되는 과정에 대한 설명은 부족하기 때문이다. 또한, 개인 정보의 사용이 어떠한 보안 환경을 통해 안전하게 제공되는지에 대한 정보와 어떠한 문제점을 내제하고 있는지에 대해서도 제공되지 않는다. 따라서 이에 대한 이해와 분석을 통해 근본적인 서비스의 이해를 돕는 것이 가능하다.

이에 따라 본 논문에서 분석하고 있는 SET는 사용자의 구매정보와 지불정보를 각각 분리하여 전송함으로써 판매자와 금융기관이 서로 자신이 필요한 정보 이외의 정보를 알 수 없게 하는 기술이다. 즉, 판매자는 사용자가 선택한 제품에 대한 구매정보만 습득이 가능하며, 금융기관은 판매자에게 지불할 금액과 사용자 신용카드 정보 등의 지불정보만 습득이 가능하다. 이와 같은 기술은 이중서명을 통해 수행되며, 전자봉투 시스템을 함께하여 기밀성까지 보장하고 있다.

그러나 SET를 수행하기 위해서 사용되는 키가 많고 암호화를 위해 사용되는 대칭키의 분배 등이 일반적인 네트워크 환경을 통해 이루어지므로, 키의 관리 및 분배 등의 안전성을 확실히 보장하기 어렵다. 또한, 다수의 서명 정보 및 해쉬값을 포함하여 보내야 하므로 전송되는 데이터의 크기도 늘어난다.

따라서 본 논문에서는 이와 같은 키 관리 및 분배의 문제와 경량화를 위하여 프록시 재암호화를 이용한 전자지불시스템을 제안한다. 또한 이에 대한 분석을 통해 사용자의 서비스 이해도를 향상시킬 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 SET와 프록시 재암호화 기법에 대한 분석을 수행하고, 3장에서는 SET의 문제점을 분석한다. 4장에서는 본 논문의 제안 기법인 안전한 전자지불시스템을 제안하고, 5장에서는 안전성 및 효율성을 분석한다. 마지막으로 6장을 결론으로 끝을 맺는다.

2. 관련연구

2.1 SET(Secure Electronic Transaction)

SET는 1997년 5월 VISA와 MASTER 카드사가 안전한 신용카드 기반의 전자상거래를 구축하기 위해서 제안한 전자지불시스템 표준 프로토콜이다. 이를 정의하면 전자상거래에서 지불정보를 안전하고 비용 효과적으로 처리할 수 있도록 규정한 표준 프로토콜이라 할 수 있다[1].

2.1.1 SET 구성

SET를 기반으로 하는 전자지불시스템은 구매자 소프트웨어, 판매자 서버시스템, 지불게이트웨이, 인증 서버시스템의 4가지 요소로 구성된다.

1) 구매자 소프트웨어

구매자 소프트웨어는 지불처리시 서버에서 전송되는 구동 메시지에 의해 일반적으로 시작되는 일종의 전자지갑 소프트웨어이다. 따라서 구매자의 구매 및 지불정보의 보호, 인증서 및 개인키 관리, 거래 내역의 관리, 판매자 서버와 연동한 SET 전자지불처리를 주요 기능으로 한다.

2) 판매자 서버시스템

판매자 서버는 구매자가 요구한 구매 및 지불 명령을 처리하는 역할을 담당한다. 이 서버시스템의 주요 기능은 거래 정보의 보호와 거래 내역을 관리하는 것이며, 또한 판매자의 공개키 및 개인키를 관리하고 고객의 주문 처리 및 지불게이트웨이에 대한 인가, 요구 및 대금 이체 등을 포함한다.

3) 지불게이트웨이

지불게이트웨이는 일종의 지불 대행 시스템으로, 판매자와 계약을 체결한 카드결제 담당 금융기관 또는 믿을 수 있는 제 3의 기관에서 운영될 수 있다. 주요 기능으로는 판매자로부터의 인가 메시지 및 대금 이체 메시지를 신용카드 결제 담당 금융기관 또는 제 3의 기관에게 전송하여 처리하는 것이다. 그 외에도 거래내역 관리, 인증서버의 인증서 취소 리스트 연동, 판매자 서버와의 연동 등을 포함한다.

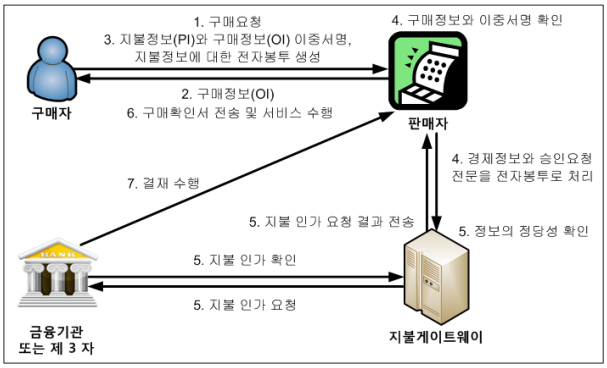
4) 인증 서버시스템

인증 서버시스템은 SET 거래에 참여하는 구성요소들에 대한 등록 및 인증서 발행을 주목적으로 수행하고 있다. 따라서 인증 서버시스템을 운영하는 인증기관은 믿을 수 있는 기관에 의해 운영되는 방식이어야 한다. 주요 기능에는 인증서의 발행 및 관리, 인증서 취소 리스트 운용, 인증서 취소 또는 재발행, 다른 인증 서버와의 연동 등이 있다.

2.1.2 SET 처리과정

SET의 전자결제 처리 과정은 다음과 같다 [2][3].

① 구매자가 판매자의 구매 목록에서 구매할 상품을 선택한다.
② 상품에 대한 가격, 종류 등의 구매정보를 판매자로부터 받는다.
③ 신용카드번호, 유효기간 등의 지불정보와 구매정보를 이중서명하고 지불 정보에 관한 전자봉투를 생성하여 판매자에게 전송한다.
④ 판매자가 구매정보와 이중서명을 확인한 후 암호화된 결제정보와 자신의 개인키로 전자서명된 승인요청 전문을 전자봉투로 처리하여 지불게이트웨이로 전송한다.
⑤ 지불게이트웨이는 전송받은 정보를 복호화하여 정당한 요청인지 확인 후, 금융기관 등과 연동하여 지불인가를 요청하고 결과를 판매자에게 전송한다.
⑥ 판매자는 지불인가가 이루어지면 구매확인서를 구매자에게 전송하고 서비스를 수행한다.
⑦ 판매자는 지불게이트웨이에 결재를 요구하고 지불게이트웨이와 금융기관 등이 연동하여 결재 요구를 수행한다.

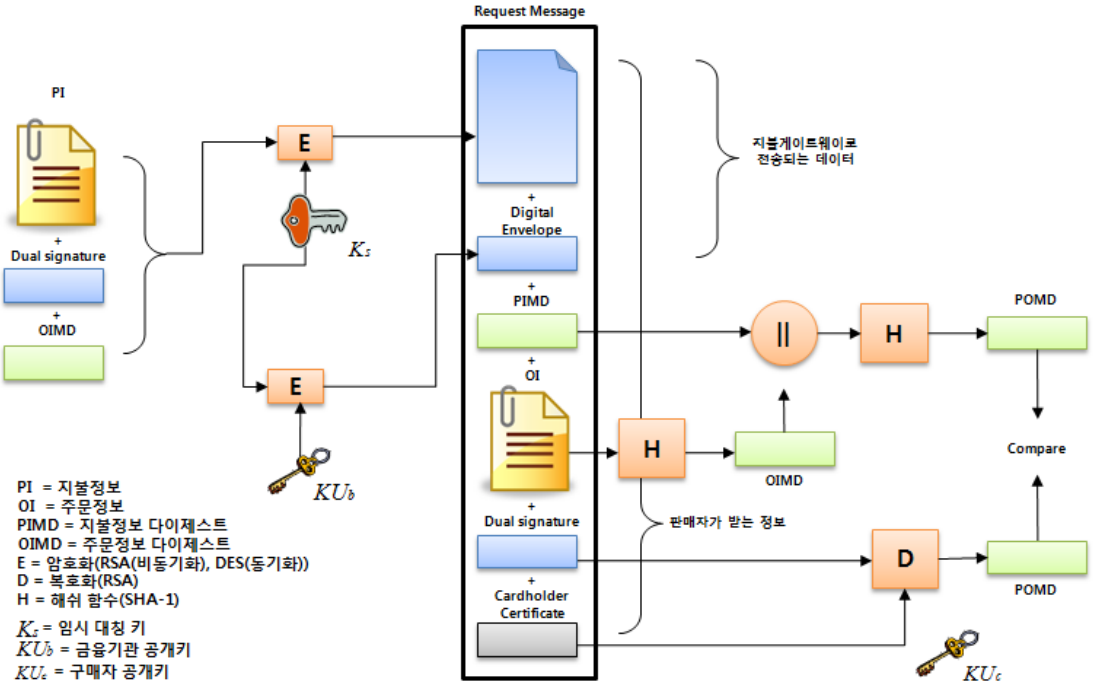


<그림 1> SET 처리 과정

2.1.3 전자봉투와 이중서명

SET가 수행되기 위해서는 전자봉투와 이중서명이 필수적으로 사용된다. 전자봉투는 SET 처리 과정에서 사용되는 전자서명이 메시지에 대한 기밀성을 보장하지 못하기 때문에 이를 해결하기 위한 방법으로 사용되는 기술이다. 본 기술은 전자서명된 메시지를 관용 암호방식으로 암호화한 후, 사용된 대칭키를 수신자의 공개키로 암호화해서 함께 전송하는 방식이다. 이를 통해 메시지의 기밀성과 전자서명을 통한 사용자 인증을 동시에 만족할 수 있다.

이중서명은 구매자가 판매자에게는 구매정보만을 보여주고, 결제기관에서는 지불정보만을 알 수 있도록 하여 각각의 투명성을 제공하기 위한 방



<그림 2> 전자봉투와 이중서명 처리도

법이다. 각 정보를 서로 다른 공개키로 암호화하고 이를 연결하여 전송하면, 판매자와 결제기관에서는 자신의 비밀키로 복호화 가능한 정보만 취득하게 되어 타 정보에 대한 안전성을 보장한다.

SET에서 사용되는 전자봉투와 이중서명 방식은 공개키 방식과 대칭키 방식을 모두 사용하고 있다. 먼저 지불정보(PI)와 주문정보(OI)를 해쉬하여 메시지 다이제스트(PIMD, OIMD)를 생성하고 이를 연접한 후 다시 해쉬를 수행하여 지불·주문 메시지 다이제스트(POMD)를 생성한다. 여기에 개인키로 서명을 하여 이중서명값을 만든다.

이 정보를 기반으로 구매자는 PI와 이중서명(dual signature), POMD를 연접하고 랜덤하게 생성한 대칭키로 암호화를 수행한다. 그리고 대칭키는 금융기관의 공개키로 암호화하여 전자봉투를 생성하고, POMD와 연접한다. 여기에 PIMD를 연접하고 OI, 전자서명, 구매자의 인증서를 연접하여 판매자에게 전송한다. 판매자는 전송받은 정보에서 OI를 해쉬하여 OIMD를 만들고 이를 PIMD와 함께 해쉬하여 POMD를 생성한다. 이렇게 생성된 POMD와 구매자의 인증서에서 공개키를 이용하여 이중서명 정보를 복호화해 POMD를 찾아내고 이를 비교하여 정당한 요청인지에 대한 정보와 사용자의 대한 인증까지 완료한다. 그 후 지불정보에 관련된 정보만을 금융기관에 보내어 지불 요청을 처리한다[4].

2.2 ebXML(electronic business XML)

ebXML은 지난 1999년 11월부터 시작되어 18개월 동안 UN/CEFACT와 OASIS가 주축이 되어 만들어낸 표준이다. 본 기술은 XML을 이용한 인터넷 기반의 전자상거래가 가능하도록 하기 위한 표준으로 설계되었다. 본 표준의 주요 특징으로는 단순한 비즈니스 문서 교환을 능가하는 프로세스 모델로서의 역할을 수행한다. 또한, 개별 사업자 간에 동일한 의미로 자주 사용되는 요소들을 핵심적인 구성 요소로 정의한 후 자신이 실제 필요한 부분에 독자적인 구축을 가능하게 하며, 저장소의 개별적인 운영이 가능한 특징이 있다[5].

2.2.1 구성요소

ebXML의 구성요소는 BP(Business Process), CC(Core Components), RR(Registry/Repository), TP(Trading Partner), MS(Message Service)의 5가지로 분류된다[6].

1) BP

당사자의 거래 절차로써 당사자의 거래업무 처리 절차에 관련된 정보들을 표준 규격에 맞추어 작성한 XML 문서이다. 해당 문서에는 기업간 거래에 있어서 공유하는 역할 및 관계, 의무사항 등에 대한 처리 방법에 대하여 정의한다. 또한 비즈니스 프로세스와 이와 관련된 정보 모델 등의 일관된 모델링 기법을 제공한다.

2) CC

전자상거래에서 상호 교환되는 전자적 메시지들을 구성하는 기본 단위들을 공유하고 이를 재사용할 수 있도록 표준화한 기능으로, 비즈니스 프로세스를 기반으로 재사용성, 확장성, 상속성을 지닌 의미의 중립적인 객체를 의미한다.

3) RR

거래에 필요한 각종 정보들을 저장 및 공유하고 표준화된 접근 방식을 적용한 기본적 정보모델을 제공한다. 여기에는 두 가지 데이터 보관에 관한 내용이 포함되는데 Registry는 서비스의 메타데이터 등에 대한 색인정보를 보관하며, Repository는 거래 상대자가 제출한 정보를 보관에 대한 내용이 포함된다.

4) TP

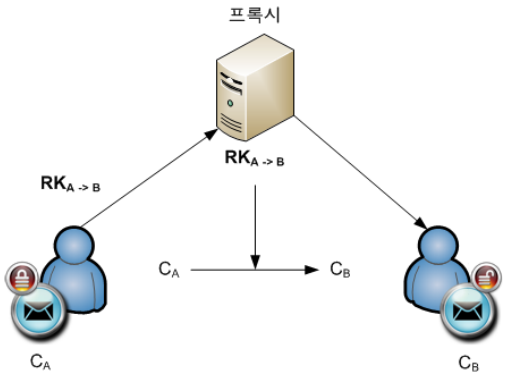
거래 당사자에 대한 정보(CPP : Collaboration Protocol Profile)와 협업을 위한 프로파일 틀(CPA : Collaboration Protocol Agreement))을 표준화하는 부분이다. 여기에서 CPP는 '협업 규약 프로파일'로써 기업간 거래를 위한 시스템 환경과 전자 거래 절차에 관한 정보를 담고 있는 문서이다. 그리고 CPA는 '협업 규약 약정서'로써 상호 협의를 통하여 공식적인 기업간 거래가 발생할 때 활용되는 약정서 역할의 문서이다.

5) MS

전자상거래에서 사용되는 각각의 요소 사이에 메시지 전송 및 보안성을 규정하는 부분으로써, 거래 당사자들간의 비즈니스 메시지들을 교환하기 위한 표준 기법을 제공한다.

2.3 프록시 재암호화(Proxy Re-encryption)

프록시 재암호화란 프록시가 A의 공개키로 암호화된 암호문을 B의 비밀키로 복호화 할 수 있도록 암호문을 복호하지 않고 변환하는 방식이다. 여기에서 프록시는 암호문을 변환하기 위한 키를 사용하여 기존의 암호문을 복호화 없이 변환하기 때문에 평문이나 A의 비밀키를 알지 못한다. 따라서 암호화된 데이터의 안전성을 보장할 수 있다.



<그림 3> 메일 전송 예

예를 들어, 메일 전송에서는 A의 부재 또는 비밀키 분실 시, A의 암호 메일을 프록시가 B의 암호 메일로 변환하여 메일을 전송하는 것이 가능하다. 프록시가 암호문을 복호하지 않고 B의 암호문으로 변환을 수행하여 전송하면, B도 A의 비밀키를 이용하지 않고 자신의 비밀키로 복호화가 가능하다. 이와 같은 방식은 B에서 A로 전송될 때에도 마찬가지로 적용된다[7].

2.3.1 프록시 재암호화 연구 동향

최초로 암호문을 변환하여 상대방이 복호화를 할 수 있는 위임 방식은 1997년 M. Mambo, E. Okamoto의 "Proxy cryptosystems: Delegation of the power to decrypt ciphertext"에서 처음으로 제안되었다[8]. 그러나 최초 제안된 방식은 A의 암호문을 A만이 변환할 수 있기 때문에 A가 부재인 경우에는 변환 자체가 불가능한 문제점이

있었다.

이와 같은 문제점을 해결하기 위하여 다양한 제안들이 시도되었다. 그 중, 1998년에 Blaze, Bleumer, Strauss가 'Atomic Proxy Cryptography' 방식을 제안하였다[9]. 이 방식은 암호문을 변환하는 프록시가 재암호화 키를 이용하여 변환을 수행한다. 따라서 A가 변환을 수행하지 않고 프록시가 이를 수행함으로써 A가 부재하더라도 변환이 가능하다는 이점을 가지게 되었다. 그러나 여기에서 사용되는 재암호화 키는 A와 B의 비밀키로부터 생성되면서, 프록시와 사용자 한명이 공모를 하게 되면 다른 사용자의 비밀키를 알아낼 수 있다는 문제점이 제기되었다. 그리고 각 사용자의 비밀키를 프록시가 이용하므로 신뢰성에 대한 문제도 발생되었다.

이를 개선한 방안이 2003년 Dodis와 Ivan에 의해 제안되었다[10]. 제안된 방식은 'Unidirectional Proxy Encryption'으로 A의 비밀키를 2개로 분할하여 하나는 프록시에 전달하여 암호문 변환에 이용하고 나머지 하나는 B에 전달하여 변환한 암호문의 복호에 이용하는 방식이다. 그러나 키를 분할하여 전송함으로써 키 분배의 문제가 발생되었다. 분리된 비밀키를 안전하게 전송하는 것이 쉽지 않다는 것이다. 또한 여전히 프록시와 B의 결탁으로 A의 비밀키가 노출될 수 있다는 문제점이 발생되었다. 위 제안 방식들의 문제점을 해결하고 최근까지 가장 효율적인 방식으로 인식되어 사용되고 있는 방법은 Ateniese, Fu, Green, Hohenberger 가 제안한 'Unidirectional Proxy Re-encryption'이다.

2.3.2 Unidirectional Proxy Re-encryption

Ateniese 등이 2005년 제안한 Unidirectional Proxy Re-encryption은 재암호를 위해 사용되는 키를 사용자의 비밀키가 아니라 송신자 A의 비밀키와 수신자 B의 공개키를 이용하여 생성한다. 또한 이와 같은 키는 A 자신이 생성할 수 있게 되면서 프록시가 이전과 같이 사용자의 비밀키를 알 수 없게 되었다[11].

이 방식은 키 생성, 암호화, 복호화(A), 재암호화, 복호화(B) 5가지의 알고리즘으로 구성된다.

<표 1> Unidirectional Proxy Re-encryption의 5가지 알고리즘

단계	알고리즘	내용
1	키 생성	- 비밀키 $SK_a = a \in Z_q^*$, $SK_b = b \in Z_a^*$ 생성 - 공개키 $PK_a = g^a$, $PK_b = g^b$ 생성 - 임의의 난수 $r \in Z_q^*$ 생성 - $Z = e(g, g)$ - 재암호화키 $RK_{A \to B} = (g^b)^{1/a} = g^{b/a}$ 생성
2	암호화	- 평문 : $m \in G_2$ - 공개키(PK_a)로 암호화 : $C_a = (Z^r \bullet m, g^{ra})$
3	복호화	- 암호문 : C_a - 비밀키(SK_a)로 복호화 : $m = \frac{Z^r \bullet m}{e(g^{ra}, g^{1/a})} = \frac{Z^r \bullet m}{Z^r}$
4	재암호화	- C_a를 프록시가 재암호화키 $RK_{A \to B}$를 이용하여 C_b로 재암호화 $C_a = (Z^r \bullet m, g^{ra})$ $C_b = (Z^r \bullet m, e(g^{ra}, RK_{A \to B}))$ $= (Z^r \bullet m, e(g^{ra}, g^{b/a}))$ $= (Z^r \bullet m, Z^{rb})$

이와 같은 방식의 재암호화 기법은 다양한 이점을 제공한다. 본 방식은 사전에 비밀키를 공유하지 않고 송신자의 비밀키와 수신자의 공개키를 사용하여 재암호화키를 생성하므로 프록시와 수신자는 송신자의 비밀키를 알 수 없다. 이는 역으로 수행해도 마찬가지이므로 자신의 비밀키가 노출되는 위험을 줄일 수 있다. 그리고 단방향성을 제공하므로 A가 생성한 키를 B가 역으로 재사용할 수 없으며, 프록시의 경우는 로부터 를 계산해 낼 수 없으므로 재암호화기능 외에 아무런 권한이 주어지지 않는다. 이와 같은 이유로 프록시와 수신자가 공모를 하더라도 송신자의 비밀키를 알아낼 수 없다[11].

3. 문제점 분석

3.1 키 분배 및 관리 문제

SET에서 사용되는 이중서명 방식은 공개키 방식과 대칭키 방식을 모두 사용하고 있다. 구매자는 구매 정보와 지불 정보에 대한 해쉬값을 각각

구하고, 이 두 개의 해쉬값을 묶은 값에 대해 또 한 번 해쉬를 하고 그 값을 서명함으로써 구매 정보와 지불 정보를 서로 연관시켜 주는 방법이다. 마지막으로 이 최종 해쉬값과 지불 정보의 해쉬값 및 구매 정보의 해쉬값에 대해 서명을 한 번 더 해준 후에 상인에게 전달한다. 상인은 이 정보들로부터 구매 정보는 복구하여 볼 수 있지만 지불 정보는 보지 못하고 단지 두 정보 사이의 연관성만을 확인할 수 있다. 마찬가지로 지불 게이트웨이는 지불 정보는 해독하여 볼 수 있지만 구매 정보는 볼 수 없고 단지 두 정보 사이의 연관성만을 확인할 수 있다.

이 때, 지불정보를 암/복호화하기 위한 키로 대칭키(K_s)를 사용하고, 이를 안전하게 분배하기 위하여 공개키(KU_b)를 사용한다. 따라서 판매자와 키를 반드시 공유해야 하므로 네트워크를 통해 데이터가 전송되어야 한다. 그러나 공개된 네트워크를 통해 전송되는 정보는 암호화를 수행하였다고 해도 완전한 안전성을 보장 받을 수는 없다. 만약 공격자가 공개키에 대응되는 개인키를 알아냈을 경우, 해당 대칭키(K_s)까지 노출되게 되어 지불 정보에 대한 직접적인 피해가 발생할 수 있는 위험이 있다.

또한, SET를 수행하기 위해서는 모든 거래에 랜덤한 대칭키를 하나씩 사용하므로 공개키만을 이용하는 방식에 비해 키의 소요가 많다. 이러한 부분은 모든 키에 대한 관리가 요구된다는 어려움이 존재한다. 이와 같은 문제점은 안전성의 하락과 효율성의 문제를 발생시킨다.

3.2 지불 정보 위조 가능성

지불정보는 사용자가 구매한 제품에 대한 금전적인 거래를 위한 정보로 위조될 경우, 금전적인 피해로 이어질 수 있는 민감한 정보가 포함된다. 그러나 SET에서는 지불정보를 암호화한 키를 같이 전송함으로써 일반적인 네트워크에 키가 노출될 수 있는 문제점을 가지고 있다. 이러한 대칭키(K_s)가 공개키(KU_b)를 통해 암호화되어 있지만, 공격자가 공개키(KU_b)의 대응되는 개인키(KP_b)를 알아내게 되면 키를 복호화하여 알아낼 수 있게 된다. 공격자는 이를 통하여 사용자의 지불정보를 획득할 수 있으며, 위조한 지불 정보를 재암호화하여 전송함으로써 금전적 이득을 취할 수 있는 가능성이 존재한다.

$$KP_b(KU_b(K_s)) = K_s$$
$$PIMD_a' = H(PI_a')$$
$$Dual\,signature = E_{K_a}(H(PIMD_a' \| OIMD))$$
$$\text{데이터 위조} : K_s(PI_a' \| Dual\,signature \| OIMD)$$

KP_b : KUb의 대응되는 개인키
PI_a' : 공격자가 위조한 지불정보
$PIMD_a'$: PIa′의 메시지 다이제스트
K_a : 공격자의 개인키

3.3 지불 정보 위조 가능성

SET에서는 정보의 보호를 위해 다수의 해쉬와 암호화를 이용하여 데이터를 처리하고 있다. 가벼운 해쉬를 사용하고 있다고는 하나 지불정보와 주문정보를 모두 해쉬하고 이것을 또다시 해쉬하여 이중서명을 얻고 이를 다시 암호화하는 등의 여러 연산과정을 거친다. 이와 같은 다수의 처리 과정은 전자결제를 수행하는데 있어 시간적인 손실을 가져오게 되며, 전체적인 시스템 성능 하락으로 이어지게 된다.

그리고 각 정보의 연관성 및 정당성 확인을 위하여 생성된 모든 정보가 전송 시에 포함되어야 하므로 단순 정보 전송에 비해 용량이 증가할 수 밖에 없다. 대용량의 정보가 발생되는 것은 아니지만 최근 이슈가 되고 있는 DDoS 공격 등에 활용된다면 전송되는 데이터 크기의 감소는 네트워크 트래픽을 감소시킬 수 있는 요소가 될 수 있다. 또한, 다수의 데이터가 전송되면서 이를 검증하기 위하여 판매자 역시 다수의 처리 과정을 수행해야하는 등의 문제가 발생한다.

4. 프록시 재암호화 기반의 안전한 전자지불시스템

본 논문에서는 기존 SET 기반의 전자지불시스템의 키 분배 및 관리 문제를 해결하고 전송되는 데이터를 경량화하기 위하여 프록시 재암호화를 이용하여 전자지불시스템을 구성하는 방안을 제

안한다.

기존의 SET 시스템은 이중서명과 전자봉투를 이용하여 데이터를 전송하였다. 따라서 이중서명을 만들기 위하여 PI와 OI에 대한 해쉬값을 만들고 이를 다시 연접하여 POMD를 생성하여 이중서명을 수행하였다. 그리고 PI와 이중서명, OIMD를 암호화하기 위한 대칭키를 생성하고 공유하기 위하여 이를 다시 금융기관의 공개키로 암호화해야 한다. 그러나 이와 같은 과정은 다수의 해쉬연산으로 인한 시간 지연과 대칭키 생성, 대칭키 공유를 위한 암호화 과정이 추가되어야 한다. 또한 일반적인 네트워크 환경으로 데이터가 전송되면서 발생할 수 있는 키의 노출 등의 문제 발생도 가능하다.

본 논문에서는 이를 해결하기 위해 이중서명과 전자봉투를 사용하지 않고 프록시 재암호화를 통해 대칭키의 공유 없이 판매자와 금융기관이 각자의 정보만 확인할 수 있도록 하는 방안을 제안한다.

<그림 4>는 제안하는 시스템의 구성도를 나타낸다.

제안 방식에서 사용되는 키는 Unidirectional Proxy Re-encryption의 5가지 알고리즘의 키 생성 알고리즘을 만족하도록 다음과 같이 구성된다.

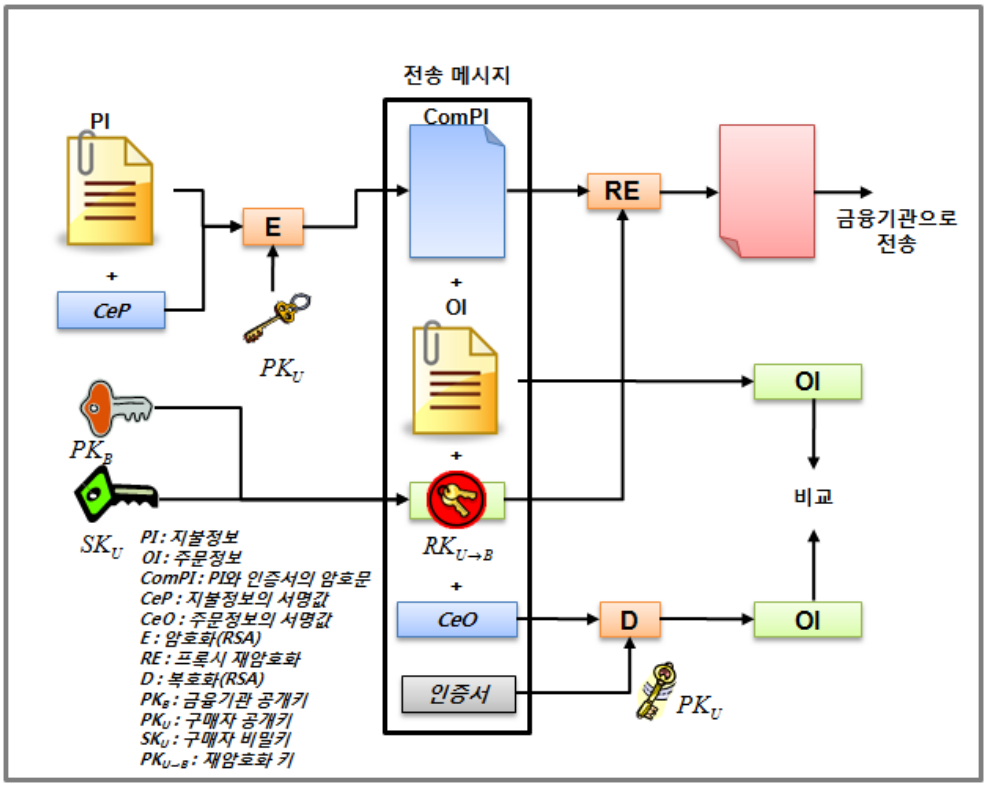
<표 2> 제안방식의 키 생성 알고리즘 구성

알고리즘	내용
키 생성	- 비밀키 $SK_U = U \in Z_q^*$, $SK_B = B \in Z_q^*$ 생성 - 공개키 $PK_U = g^U$, $PK_B = g^B$ 생성 - 임의의 난수 $r \in Z_q^*$ 생성 - $Z = e(g, g)$ - 재암호화키 $RK_{U \to B} = (g^B)^{1/U} = g^{B/U}$ 생성

제안하는 방안은 구매자, 판매자, 금융기관의 세 부분으로 나누어 구성된다.

4.1 구매자 과정

구매자는 자신이 원하는 제품을 구매하기 위해 물품을 선택하여 OI와 PI를 구성한다. 이는 기존 SET 방식과 동일하다. 이 후, PI는 사용자의 개인키(SK_U)를 이용하여 서명값(CeP)을 생성하고 다시 사용자의 공개키(PK_U)로 암호화를 수행하여



<그림 4> 제안하는 시스템 구성도

구매자 PI와 서명값의 암호문($ComPI_U$)을 생성한다. 이와 같이 생성된 정보는 재암호화 전에는 구매자만이 복호화가 가능하므로 외부로 노출되어도 공개키 기반의 안전성을 보장한다. 또한 CeP는 지불정보의 위/변조를 확인하기 위하여 사용된다.

$$CeP = E_{SK_U}(PI)$$
$$m = PI \| CeP$$
$$ComPI_U = (Z^r \bullet m, PK_U{}^r)$$
$$= (Z^r \bullet m, g^{rU})$$

주문정보 또한 개인키로 서명하여 서명값을 생성하고 OI와 주문정보 서명값(CeO), $ComPI_U$를 연접한다. 주문정보의 서명값은 주문정보의 위/변조를 확인하기 위하여 사용된다.

$$CeO = E_{SK_U}(OI)$$
$$Message = ComPI_U \| OI \| CeO$$

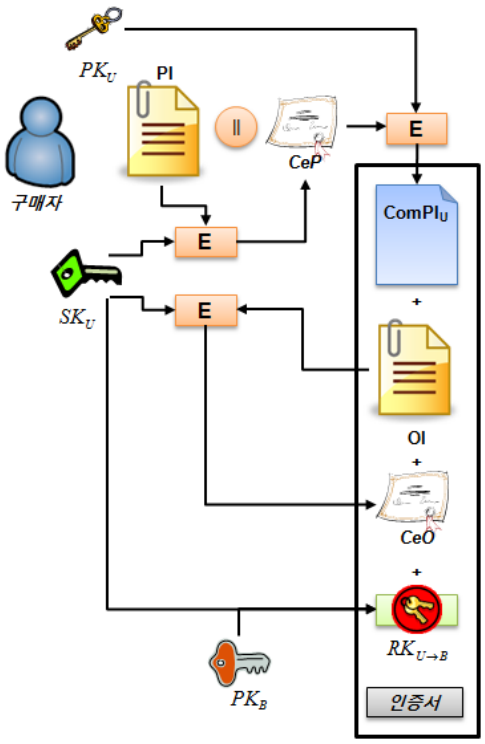
구매자는 자신의 비밀키와 금융기관의 공개키를 이용하여 프록시 재암화를 위한 키를 생성한다. 해당 키는 판매자가 금융기관에 $ComPI_U$를 전송하기 전에 재암호를 위해 사용한다. 재암호화 키를 통해 복호화 과정 없이 금융기관이 복호화할 수 있도록 변환할 수 있다. Unidirectional Proxy Re-encryption의 5가지 알고리즘의 키 생성 알고리즘을 만족하므로 다음과 같이 사용된다.

$$RK_{U \to B} = g^B \bullet g^{1/U}$$
$$= (g^B)^{1/U}$$
$$= g^{B/U}$$

마지막으로 각 정보와 인증서를 합쳐 판매자의 공개키로 암호화한 후 전송한다.

구매자 → 판매자 :

$$Message = E_{PK_M}(ComPI \| OI \| CeO \| RK_{U \to B} \| Certificate)$$



<그림 5> 구매자 과정

4.2 판매자 과정

판매자는 전송받은 정보 중에서 전송된 정보의 무결성을 검증하고 데이터의 위변조가 발생되지 않았다면 주문 처리를 수행한다.

먼저 전송받은 정보를 판매자의 개인키로 복호화하고 사용자의 인증서에서 취득한 공개키를 이용하여 CeO를 복호화 한다. 그 후 이를 OI와 비교하여 무결성 여부를 확인한다. 또한 인증서를 통하여 정당한 사용자인지에 대한 검증도 동시에 수행된다.

$$D_{SK_M}(E_{PK_M}(ComPI \| OI \| CeO \| RK_{U \to B} \| Certificate))$$
$$Certificate \Rightarrow PK_U$$
$$OI' = D_{PK_U}(CeO) = D_{PK_U}(E_{SK_U}(OI))$$
$$compare,$$
$$OI = ? OI'$$

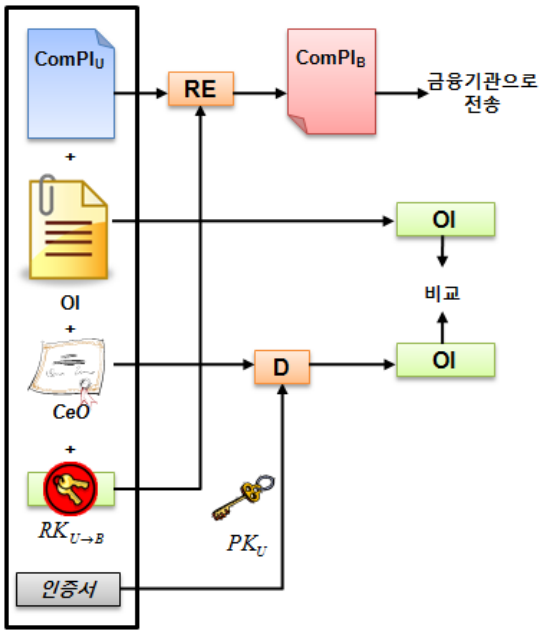
OI의 무결성이 검증되면 구매자가 구매한 제품의 발송을 준비하고 지불정보를 처리하기 위해 재암호화를 수행한다. 판매자는 재암호화키를 통해 $ComPI_U$를 금융기관의 비밀키로만 풀 수 있는

$ComPI_B$로 변환한다. 이 때, 데이터의 복호화 과정을 거치지 않으므로 데이터의 기밀성을 보장할 수 있으며, 재암호화키는 사용자의 비밀키로 암호화된 데이터만 금융기관의 데이터로 변경할 수 있기 때문에 다른 데이터를 임의로 변경하는 것이 불가능하다. 또한, 재암호화 키는 구매자의 비밀키를 이용하여 만들어졌으므로, 값을 변경할 수 없다. 이를 알아내기 위해서는 사용자의 비밀키를 반드시 알아야하므로 공개키 기반의 안전성을 보장한다. 생성된 정보는 금융기관으로 전송된다.

$$
\begin{aligned}
ComPI_U &= (Z^r \bullet PI,\ g^{rU}) \\
ComPI_B &= (Z^r \bullet PI,\ e(g^{rU}, RK_{U\to B})) \\
&= (Z^r \bullet PI,\ e(g^{rU}, g^{B/U})) \\
&= (Z^r \bullet PI,\ Z^{rB})
\end{aligned}
$$

4.3 금융기관 과정

금융기관은 전송된 정보를 개인키를 이용하여 복호화하고 구매자의 서명정보를 검증한다. 해당 서명정보를 통해 데이터의 무결성을 확인하고 구매자의 인증을 완료하면 해당 지불 결제를 위한 과정을 수행한다. 지불 결제는 기존 SET의 방식과 동일하다. 해당 부분은 단순 복호화 및 결제 수행이기 때문에 상세한 내용은 생략한다.



<그림 6> 판매자 과정

5. 안전성 및 효율성 분석

본 논문에서는 기존 SET의 키 분배 및 관리와 많은 연산량 등의 문제점을 해결하기 위해 프록시 재암호화 기법을 이용한 전자지불시스템을 제안하였다. 본 장에서는 기존 SET와의 안전성 및 효율성을 비교 분석한다.

5.1 키 분배 및 관리 문제

기존의 SET 방식은 키의 분배 및 관리를 위하여 공개키와 대칭키 방식을 모두 사용하고 있다. 지불정보를 암/복호화하기 위한 키로 대칭키(K_s)를 사용하고, 이를 안전하게 분배하기 위하여 공개키(KU_b)를 사용한다. 따라서 판매자와 키를 반드시 공유해야 하므로 네트워크를 통해 데이터가 전송되어야 한다. 이는 키가 전송되지 않는 시스템에 비해 상대적인 위험성을 가지고 있다. 만약 공격자가 공개키에 대응되는 개인키(KP_b)를 알았을 경우, 해당 대칭키(K_s)까지 노출되게 되며, 이를 통해 지불정보(PI_a') 자체에 대한 위조가 가능해져 금전적인 손실이 발생할 수 있다.

그러나 본 논문에서 제안한 기법은 전송되는 메시지에 지불정보를 암호화하는데 사용된 키를 포함하지 않으며, 사용하는 키가 구매자의 개인키이기 때문에 더욱 외부로 유출될 위험이 적다. 또한, 공개키 방식만 사용하여 대칭키를 사용함에 있어 키의 분배과정이 필요 없다. 이는 데이터 노출의 위험을 줄이고 키의 안전성을 더욱 높일 수 있다.

5.2 지불정보의 위조

지불정보를 암호화하는 방식은 기존 SET는 대칭키를 이용하고 있으며, 본 제안 방안은 공개키 방식을 기반으로 하고 있다. 안전성을 기준으로 분석하였을 때, 대칭키 암호 방식에 비해 상대적으로 공개키 암호 방식이 공격에 안전하다. 상대적으로 처리하는데 걸리는 시간은 좀 더 오래 걸릴 수 있으나, 지불정보의 크기가 그다지 크지 않으므로 이에 걸리는 시간 차이는 미비할 것으로 분석된다. 대칭키 암호화 방식을 이용하는 기존

SET는 대칭키가 전송되는 정보에 포함되어 전송된다. 이를 복호화할 경우, 지불정보에 대한 복호화가 가능해지고, 자신이 원하는 키로 다시 암호화하여 전송하는 등의 문제가 발생할 수 있다.

그러나 본 논문의 제안방식은 지불정보를 복호화할 수 있는 정보를 전송되는 메시지에 포함하지 않으므로, 지불정보의 복호화가 불가능하다. 공격자가 사용자의 개인키를 알아야만 해당 정보를 볼 수 있기 때문에, 공개키 기반의 안전성을 보장한다. 또한, 공격자가 임의의 위조된 정보를 생성하여 전송하더라도 재암호화를 위한 키를 동일하게 생성할 수 없다. 재암호화키는 사용자의 개인키(SK_U)와 금융기관의 공개키(PK_B)를 이용하여 생성한다. 재암호화키를 임의로 생성하기 위해서는 주문정보(OI)의 서명값(CeO)를 검증하기 위해서 포함된 인증서까지 변경해야 한다. 그러나 이를 변경하면, 정당한 사용자의 식별정보와 다른 정보가 포함되어 확인이 가능하게 되므로, 이와 같은 위조가 불가능하다.

5.3 연산량

연산량을 비교하여 보면, SET 방식은 4번의 암호화와 5번의 해쉬함수가 사용되고, 세션키 생성을 위한 키 생성 알고리즘이 반드시 수행되어야 한다. 이에 반해 본 논문에서 제안하고 있는 방식은 5번의 암호화 과정만 수행된다. 암호화 과정이 해쉬함수 및 키 생성 알고리즘에 비해 느린 것은 분명하나 5번의 해쉬함수와 1번의 키 생성 알고리즘을 모두 합치면 본 논문에 추가된 1번의 암호화 과정의 시간을 넘을 것으로 분석된다. 이는 전자상거래를 위해 사용되는 데이터가 일반적인 데이터에 비해 상대적으로 작아 암호화에 걸리는 시간이 비교적 짧기 때문이다. 따라서 본 논문의 연산량이 기존 SET의 연산량에 비해 추가로 소요되지는 않다고 분석된다.

5.4 전송 데이터 용량

모든 과정이 완료되고 실제 전송되는 데이터의 용량을 보면, SET에 비해 본 제안 방안이 좀 더 적다는 것을 알 수 있다. SET는 총 6 종류의 데이터를 연접하여 전송하고, 제안 방식은 총 5 종류의 데이터를 연접하여 전송한다. 본 제안 방식에서 전송되는 정보가 기존의 SET 방식에서 전송되는 정보와 크게 다르지 않기 때문에 유사한 부분을 모두 제거해도 SET 방식에서는 PIMD가 남아있다. 이는 본 제안 방식이 SET 방식에 비해 전송되는 데이터를 경량화 할 수 있다는 것을 의미한다.

<표 3> 안전성 및 효율성 분석

	기존 SET 방식	제안 방식
키 분배 및 관리 문제	- 공개키 방식과 대칭키 방식 모두 사용 - 대칭키 분배에 대한 위험 존재 - 대칭키 노출 시 지불정보 위조 가능 - 다수의 키 사용으로 관리 부담 증가	- 공개키 방식만 사용 - 다수의 키를 관리할 필요 없음 - 구매자의 개인키 없이 지불정보 위조 불가능
지불정보의 위조	- 금융기관의 공개키로 암호화되어 있는 대칭키를 알아야 지불정보 노출 - 지불정보 암호 방식이 공개키 방식에 비해 상대적으로 공격에 취약	- 구매자의 비밀키를 알아야 지불정보 노출(대칭키에 비해 상대적으로 안전) - 복호화를 위한 정보가 전송되지 않음 - 재암호화키 생성 시, 인증서까지 변경해야 함 (정당한 사용자 식별 가능)
연산량	(판매자까지) - 4E + 5H - 세션키 생성 과정 추가	(판매자까지) - 5E
전송 데이터 용량	- 총 6개의 데이터 포함 전송	- 총 5개의 데이터 포함 전송

6. 결론

전자금융거래가 다양한 스마트기기의 발전, 네트워크의 발전 등으로 다양한 사람들이 손쉽게 이용할 수 있게 되면서 폭발적인 성장세를 가져왔다. 현재의 전자상거래는 안정기에 접어들면서 좀 더 효율적이고 안전한 시스템에 대한 욕구가 증가하게 되었다. 이에 다양한 전자지불시스템에 대한 연구가 진행되고 있으며, 기존의 전자지불시스템에 대한 개선 방향도 제시되고 있다.

특히 SET는 신용카드 기반의 전자지불시스템 표준으로 널리 사용되고 있는 시스템이다. 그러나 이러한 시스템 또한 다수의 키를 관리하고, 키 분배에 대한 위험이 포함되어 있다. 또한 다수의 암호화 및 해쉬함수의 작동으로 연산량도 적지 않

은 편이다.

이에 본 논문에서는 기존의 전자상거래를 위한 시스템 중 SET의 문제점을 분석하고 이를 해결하기 위해 프록시 재암호화 기반의 안전한 전자지불시스템을 제안하였다. 프록시 재암호화 기법을 통하여 기존 SET의 키 분배 문제를 해결하고, 다수의 키를 관리해야 하는 문제점을 해결하였다. 또한, 연산량도 상대적으로 적게 소요되어 효율적인 통신이 가능하다.

이와 같은 제안 방식을 통하여 기존의 전자지불시스템의 효율성 및 안전성을 증대하고 안전한 전자상거래 환경을 구성할 수 있을 것이다. 또한, 전자지불시스템에 대한 전반적인 이해와 서비스의 근본적인 동작 방식에 대해 이해할 수 있으며, 보안상의 문제점을 고찰하고 안전한 시스템 제안을 통해 서비스의 나아가야할 방향에 대해 제고할 수 있을 것으로 기대된다.

참 고 문 헌

[1] Kawatsura, Y. (2003). *Secure Electronic Transaction(SET) supplement for the v1.0 Internet Open Trading Protocol(IOTP).* RFC3538.

[2] 장우석, 이광우, 최동현, 정학, 이병희, 최윤성, 김승주, 원동호 (2005). SET와 그 변형기법들에 관한 연구. **한국정보보호학회지, 15**(4), 17-28.

[3] 김진아, 박찬정 (2006). S_Set(Strong Secure Electronic Transaction)을 이용한 T-Commerce. **한국멀티미디어학회 추계학술발표대회논문집,** 724-727.

[4] 김근옥, 남정현, 김승주, 원동호 (2003). 효율적인 메시지 복호화를 제공하는 이중 전자서명 방식. **한국정보보호학회논문지, 13**(5), 130-136.

[5] Mongiello, M. (2006). Finite-state verification of the ebXML protocl. *Electronic Commerce Research and Applications, 39*(2), 147-169.

[6] 김채미, 최학열 (2001). **글로벌 e비즈니스 리더를 위한 ebXML.** 서울: 대청미디어.

[7] 구우권, 황정연, 김형중, 이동훈 (2009). CCA 안전성을 제공하는 ID기반 프락시 재암호화 기법. **전자공학회논문지, 46**(1), 64-77.

[8] Mambo M., & Okamoto E. (1997). Proxy cryptosystems: Delegation of the power to decrypt ciphertext. *IEICE Trans. Fund Electronics Communications and Computer Science, E80-A*(1), 54-63.

[9] Blaze M., Bleumer G., & Strauss M. (1998). Divertible protocols and atomic proxy cryptography. *EUROCRYPT'98, LNCS 1403*, 127-144.

[10] Dodis Y., & Ivan A. (2003). Proxy Cryptography revisited. *In Network and Distributed System Security Symposium.*

[11] Ateniese G., Fu K., Green M., Hohenberger S. (2005). Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. *In Network and Distributed System Security Symposium.*

고 웅

2008 순천향대학교 정보보호학과(공학학사)
2010 순천향대학교 정보보호학과(공학석사)
2010~현재 순천향대학교 정보보호학과 박사과정
관심분야: 암호프로토콜, 개인정보보호, 클라우드 컴퓨팅 보안, 융합 보안 등
E-Mail: wgo@sch.ac.kr

곽 진

1994~2006 성균관대학교(공학학사, 석사, 박사)
2006~2006년 일본 큐슈대학교 방문연구원
2006~2006 일본 큐슈시스템 정보기술연구소 특별연구원
2006~2007 정보통신부 개인정보보호기획단 개인정보보호팀 통신사무관
2007~2009 정보통신연구진흥원 집필위원
2007~현재 순천향대학교 정보보호학과 교수
2009~2009 순천향대학교 공과대학 교학부장
2009~2010 순천향대학교 정보보호학과 학과장
2010~2010 교육과학기술부 국가기술수준평가 전문위원
현재: 정보통신산업진흥원 기술평가위원, 사)국제정보능력평가원 쇼핑몰 플래너 자격 검정 출제 및 채점위원, 한국과학기술정보연구원 충남 과학기술 정보협의회 전문위원, 지식경제부 지식경제기술혁신평가단 평가위원, 순천향BIT 창업보육센터 센터장, 순천향대학교 중소기업산학협력센터 센터장
관심분야: 암호프로토콜, 응용시스템보안, 개인정보보호, 정보보호제품평가, 클라우드 컴퓨팅 보안 등
E-Mail: jkwak@sch.ac.kr