EMV 기반의 전자지불 PKI와 효율적인 IC 카드 인증메커니즘

송 상 헌[†] · 최 석 진[††] · 류 재 철[†††]

요 약

최근 자기띠 방식의 금융카드를 IC 카드로 대체하기 위해 현금카드를 위한 금융IC카드표준 규격, 신용카드를 위한 EMV 규격 등이 채택되어 관련 인프라 구축이 활발하게 전개되고 있는 상황이다. 본 논문에서는 공개키 암호를 채택하고 있는 EMV 규격을 분석함으로써 인터넷 PKI, WAP PKI 등에 비해 상대적으로 연구가 미진한 EMV 기반 Payment PKI에 대한 연구를 하고자 한다. 이와 함께 IC 카드 기반 전자결제 시스템 개발에 활용할 수 있는 EMV 기반 Payment PKI 모델을 제안하고, EMV CA 시스템을 개발하였다. 또한 이를 활용하여 EMV 규격에 정의된 IC 카드 인증메커니즘을 보완하여 IC 카드 메모리 낭비 감소, 거래 처리 시간 단축, 효율적인 운영환경 및 성능을 향상시킬 수 있는 "효율적인 IC 카드 인증메커니즘" 제안하고, 성능평가를 하였다.

Payment PKI based on EMV and Efficient IC Card Authentication Mechanism

Sang Heon Song[†] · Seok Jin Choi[††] · Jea Cheol Ryou[†††]

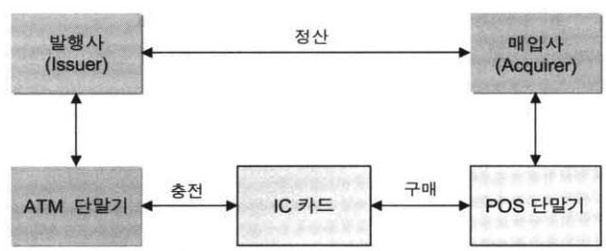
ABSTRACT

Recently "Banking IC Card Standard" and EMV Standard by the domestic standard is selected, and it is situation that is developing infrastructure vigorously to alternate Magnetic Stripe card by IC card. This paper analyzes EMV standard that is selecting public key cipher, and research wishes to study unexhausted EMV PKI relatively than internet PKI, WAP PKI etc.. This paper propose utilizable EMV base Payment PKI model in IC card base payment system development, and developed EMV CA system with this. Also, this paper supplemented IC card Authentication mechanism that is defined in EMV standard, and propose "Efficient smart card Authentication mechanism" to improve performance of this mechanism, and estimate performance.

키워드 : IC 카드(IC Card), 인증기관, 인증메커니즘(Authentication Mechanism), 전자지불(Payment System)

1. 서 론

국내외적으로 자기띠(Magnetic Stripe ; MS) 방식의 현금 및 신용카드 위변조가 심각한 사회문제로 대두되고 있다. 현재 사용하고 있는 금융카드(현금카드, 신용카드)는 자기띠에 카드정보 및 개인정보가 기록되어 있는데, 이 카드는 정보의 기록과 삭제가 용이한 매체의 특성상 보안에 매우 취약하기 때문이다. 이와 같은 문제점을 해결하기 위해 기존의 자기띠 방식의 카드를 IC 카드로 대체하기 위한 움직임이 활발하게 전개되고 있다. 그러나 IC 카드를 이용해 신용카드와 같은 전자결제시스템을 구축하기 위해서는 (그림 1)과 같이 IC 카드, ATM단말기, POS 단말기, 발행사(Issuer) 및 매입사(Acquirer) 등과 같은 은행 호스트 시스템 부분에 추가적인 인프라가 개발되어야 하는데, 이를 위해 먼저 IC 카드 관련 표준화가 반드시 선결되어야 한다.



(그림 1) IC 카드 이용 환경

※ 본 논문의 제3저자는 대학IT연구센터 육성 지원의 결과로 수행되었음.
† 정 회 원 : (주)시큐컴
†† 정 회 원 : 국가보안기술연구소 선임연구원
††† 종신회원 : 충남대학교 정보통신공학부 교수
논문접수 : 2004년 6월 8일, 심사완료 : 2004년 9월 6일

국내에서는 2003년 초 자기띠 방식의 현금카드 위변조가 심각한 사회문제로 대두된 바 있다. 이에 따라 2004년부터 점진적으로 기존의 자기띠 방식의 금융카드를 IC 카드로 대체하기 위한 "금융IC카드표준" 규격을 마련하였다[1]. 또한 국외의 마스터카드, 비자카드 등과 같은 신용카드사는 이미 1996년 경 EMV 규격을 발표한 바 있으며, 2008년까지 점진적으로 모든 신용카드를 IC 카드로 교체할 계획을 세우고 있다. 국내에서 이러한 신용카드를 널리 사용하고 있는 상황에서 EMV 규격을 국내 표준으로 수용하고 있다[2].

EMV 규격은 결제시스템의 안전성(Security)을 위해 IC 카드를 기반으로 대칭키 암호 기술과 공개키 암호 기술을 모두 수용하고 있으며, EMV 2000 Security and Key managements (Book 2) 규격에 EMV 인증서 규격과 IC 카드 인증메커니즘를 정의하고 있다. 그러나 이러한 공개키 암호 기술을 활용하기 위해서는 반드시 EMV PKI(Public Key Infrastructure)가 구축되어야 한다. 기존의 X.509 인증서 기반의 인터넷 PKI를 그대로 활용하는 방안이 있을 수 있지만, EMV 규격은 IC 카드의 저장용량의 제약, 폐쇄형 금융망 환경, 복잡한 인터넷 PKI의 구조 등으로 인해 기존의 X.509 공개키 인증서를 따르지 않고, 별도로 EMV 공개키 인증서 규격을 새롭게 정의하고 있다[3].

본 논문에서는 EMV 공개키 인증서 규격을 기반으로 EMV PKI 및 EMV IC 카드 인증메커니즘이 지니는 여러 가지 문제점을 해결하기 위한 구현모델을 제안하고, 이를 설계 및 구현을 하였다. 이와함께 기존의 EMV IC 카드 인증메커니즘(SDA, DDA)의 운영환경 및 성능을 개선할 수 있는 "효율적인 IC 카드 인증메커니즘(SDAx, DDAx)"을 제안하고, "효율적인 IC 카드 인증메커니즘"을 구현을 통해 성능평가를 하였다. 논문의 구성은 2장에서 EMV PKI와 IC 카드 인증메커니즘(SDA, DDA)을 분석하고, 3장은 본 논문에서 제안하고 있는 "EMV기반 전자지불 PKI 구현모델"과 "효율적인 IC 카드 인증메커니즘"을 살펴보고자 한다. 4장은 EMV CA 시스템을 개발하여 EMV 인증서를 생성하고, "효율적인 IC 카드 인증메커니즘"에 대한 성능을 평가하고, 5장에서 결론을 맺는다.

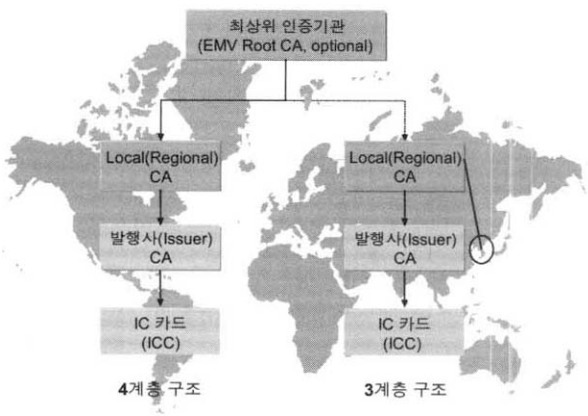
2. EMV PKI

신용카드를 IC 카드로 대체하기 위해서는 IC 카드, 카드 리더기가 부착된 ATM/POS 단말기 같은 하드웨어(H/W) 인프라 구축이 필요하다. 따라서 IC 카드 관련 인프라 구축에 필요한 표준화 작업의 일환으로 1996년 6월에 유로페이(Europay), 마스터카드(Master Card), 비자(VISA) 카드 사는 IC 카드를 이용한 신용, 직불 등과 같은 전자지불시스템 개발에 필요한 규격인 EMV 규격을 발표하였다[2-3]. ISO/IEC 7816를 기반으로 한 이 규격은 IC 카드와 단말기 H/W 규격, 보안 규격, 신용/직불과 같은 어플리케이션 규격 등이 정의되어 있다[4-6].

본 장에서는 관련연구로써 EMV 2000 Book 2 - Security and Key managements - 규격에 정의된 EMV 공개키 인증서 기반 EMV PKI 개념모델 및 IC 카드 인증메커니즘에 대한 분석을 하고자 한다. 이를 통해 본 논문에서 제안한 "EMV기반 전자지불 PKI 구현모델"을 설계하고, 기존의 EMV IC 카드 인증메커니즘(SDA, DDA)의 운영환경 및 성능을 개선할 수 있는 방안을 살펴보고자 한다.

2.1 EMV PKI 개념 모델

EMV 규격에서 정의하고 있는 EMV PKI는 (그림 2)과 같이 3계층 또는 4계층 구조를 고려하고 있다. 4계층 구조는 전 세계에 유일한 최상위인증기관을 운영하는 형태로써 선택사항으로만 언급하고 있으며, 로컬(Local) CA에 대한 인증서 규격이 정의되지 않은 상황에서 운영은 불가능한 구조이다. 3계층 구조는 로컬 CA에 대한 공개키 규격(인증서 규격이 아님)과 발행사용 EMV 공개키 인증서, IC 카드용 EMV 인증서 규격을 정의하고 있으며, 이를 운영하기 위한 모델이다[3].



(그림 2) EMV PKI 계층구조

EMV 규격에서 정의하고 있는 EMV 인증서 규격과 EMV PKI의 인증서 검증 절차, 이를 통해 운영되는 IC 카드 인증메커니즘을 살펴보면 다음과 같다.

2.2 EMV 인증서 규격

EMV 규격에서 정의하고 있는 Local Root CA의 공개키 규격(인증서 규격이 아님)과 발행사(Issuer) EMV 인증서, IC 카드 EMV 인증서 규격은 다음과 같다.

2.2.1 로컬루트(Local Root) CA 공개키

로컬루트 CA 공개키 최소 규격은 <표 1>과 같으며, 자체서명한 인증서의 형태가 아니며, 단지 공개키의 해쉬값만으로 무결성을 확인하도록 하고 있다[3].

〈표 1〉 Local Root CA 공개키 최소 규격

필 드	길 이	의 미
RID	5	Registered Application Provider Identity
Index	1	CA Public Key index
ALGH	1	해쉬 알고리즘 식별자, '01'은 SHA-1
ALGP	1	전자서명 알고리즘 식별자
LPKM	1	공개키 계수(Modulus)의 길이
LPKE	1	Exponent 길이, 1 또는 3
PKM	Var.	공개키 계수, 최대 248바이트
PKE	LPKE	Exponent
Hash	20	공개키 데이터에 대한 SHA-1해쉬값

2.2.2 발행사 EMV 인증서 규격

발행사용 EMV 인증서 규격은 <표 2>와 같으며, 로컬루트(Local Root) CA의 개인키를 이용해 IOS 9796-2(Partial Message Recovery) 방식으로 모든 필드의 값을 전자서명 한다. 이때 루트 CA의 키 길이에 따라 <표 2>의 데이터 중 일부가 암호화된 EMV 공개키 인증서 부분과 암호화에 포함되지 않는 공개키 나머지(Remainder) 값으로 구분되어 발행사용 EMV 인증서가 생성된다.

〈표 2〉 발행사 EMV 인증서 규격

필 드	길 이	설 명
Header	1	'6A'
Format code	1	0x02
ID	4	발행사(Issuer) 식별자
CED	2	인증서 유효 날짜(MM/YY)
CSN	3	CA가 부여한 인증서 일련번호
ALGH	1	해쉬 알고리즘 식별자, '01'은 SHA-1
ALGP	1	공개키 알고리즘 식별자, '01'은 RSA
LPKM	1	공개키 계수(Modulus)의 길이
LPKE	1	Exponent 길이
PKM	LPKM	공개키
PKE	LPKE	Exponent
Hash	20	인증서 데이터에 대한 해쉬값
Tailer	1	'BC'

2.2.3 IC카드 EMV 인증서 규격

IC 카드용 EMV 인증서 규격은 <표 3>과 같으며, 발행사의 개인키를 이용해 IOS 9796-2(Partial Message Recovery) 방식으로 모든 필드의 값을 전자서명 한다. 발행사 EMV 인증서 생성과 마찬가지로 발행사 CA의 키 길이에 따라 <표 3>의 데이터 중 일부가 암호화된 EMV 공개키 인증서 부분과 암호화에 포함되지 않는 공개키 나머지(Remainder) 값으로 구분되어 IC 카드용 EMV 인증서가 생성된다.

〈표 3〉 IC 카드 EMV 인증서 규격

필 드	길 이	설 명
Header	1	'6A'
Format code	1	0x04
ID	4	발행사 식별자
ICC_ID	6	ICC 식별자
CED	2	인증서 유효 날짜(MM/YY)
CSN	3	발행사 CA가 부여한 인증서 일련번호
ALGH	1	해쉬 알고리즘 식별자, '01'은 SHA-1
ALGP	1	공개키 알고리즘 식별자, '01'은 RSA
LPKM	1	공개키 계수(Modulus)의 길이
LPKE	1	Exponent 길이
PKM	LPKM	공개키
PKE	LPKE	Exponent
Hash	20	인증서 데이터에 대한 해쉬값
Tailer	1	'BC'

2.3 EMV PKI 인증서 검증

EMV 인증서의 검증은 모두 단말기에서 처리된다. 이러한 인증서 검증 절차를 크게 CA 공개키 검색 및 검증, 발행사 EMV 인증서 검증, IC 카드 EMV 인증서 검증으로 구분하여 설명하면 다음과 같다[3].

2.3.1 CA 공개키 검증

① 단말기는 <표 1>과 같은 CA공개키 리스트를 미리 저장하고 있어야 한다.
② 단말기는 IC 카드에서 로컬루트 CA 공개키 인덱스(CA Public Key Index)를 읽어온다.
③ CA 공개키 인덱스를 이용해 단말기에 저장된 CA 공개키를 검색할 수 없다면, 인증서 검증은 종료한다.
④ 로컬루트 CA 공개키 인덱스를 이용해 단말기에 저장된 CA 공개키를 검색하여 해쉬값을 비교하여 검증을 완료한다.

2.3.2 발행사 EMV 공개키 인증서 검증

① 단말기는 IC 카드에 저장된 발행사 EMV 공개키 인증서와 공개키 나머지 값(Remainder)를 읽어온다.
② "2.3.1 CA 공개키 검증"을 통해 단말기에 저장된 CA 공개키를 읽어온다.
③ 발행사 EMV 공개키 인증서의 길이가 CA의 공개키 Modulus 길이와 같은지 비교한다.
④ CA 공개키로 발행사 공개키 인증서를 CA 공개키 포멧에 정의된 알고리즘으로 복호화한다.
⑤ 복호화된 데이터(<표 2> 참조)의 Tailer가 '0xBC'와 동일한지 비교한다.
⑥ 복호화된 데이터의 Header가 '0x6A'인지 비교한다.
⑦ 인증서 포멧을 검사한다. 발행사 공개키 인증서는 '0x02'이다.

⑧ 복호화된 데이터와 공개키의 나머지 데이터를 연결하여 발행사 공개키 데이터(<표 2> 참조)를 구성한다.
⑨ 발행사 공개키 데이터내의 해쉬 알고리즘 식별자로 지정된 해쉬 알고리즘을 이용해 공개키 데이터의 해쉬값을 계산한다.
⑩ 계산한 해쉬값과 복호화된 데이터에 포함된 Hash Result값과 같은지 비교한다(공개키 인증서 변조여부를 검사).
⑪ 발행사 식별자와 PAN(leftmost 3-8 digits)값과 일치하는지 검사한다.
⑫ 공개키 인증서의 유효기간을 검사한다. 유효기간 필드의 값이 현재 날짜와 같은지 큰지를 비교한다. 현재 날짜보다 작다면 유효기간 만료로 처리한다.
⑬ RID, CA 공개키 인덱스, 인증서 일련번호가 유효한지 검사한다.
⑭ Issuer 공개키 알고리즘 식별자를 인식할 수 있는지 검사한다.
⑮ 모든 검사를 성공했다면, 발행사 EMV 공개키 인증서 검증을 완료한다.

2.3.3 IC카드 EMV 공개키 인증서 검증

① 단말기는 IC 카드에 저장된 IC 카드용 EMV 공개키 인증서와 공개키 나머지 값(Remainder)를 읽어온다.
② "2.3.1 CA 공개키 검증"을 통해 단말기에 저장된 CA 공개키를 읽어온다.
③ "2.3.2 발행사 EMV 인증서 검증"을 통해 발행사 공개키를 얻는다.
④ IC 카드 공개키 인증서의 길이가 발행사의 공개키 Modulus 길이와 같은지 비교한다.
⑤ 발행사 공개키로 IC 카드 공개키 인증서를 공개키 포멧에 정의된 알고리즘으로 복호화(recovery)한다. 복호화된 데이터의 Tailer가 '0xBC'와 동일한지 비교한다.
⑥ 복호화된 데이터의 Header가 '0x6A'인지 비교한다.
⑦ 인증서 포멧을 검사한다. IC 카드 공개키 인증서는 '0x04'이다.
⑧ 복호화된 데이터와 Remainder, Exponent를 연결하여, IC 카드 공개키 데이터를 구성한다.
⑨ 발행사 공개키 데이터 내의 해쉬 알고리즘 식별자로 지정된 해쉬 알고리즘을 이용해 공개키 데이터의 해쉬값을 계산한다.
⑩ 계산한 해쉬값과 복호화된 데이터에 포함된 Hash Result값과 같은지 비교한다(공개키 인증서 변조여부를 검사).
⑪ 복호화된 PAN(leftmost 3-8 digits)값과 Application PAN과 일치하는지 검사한다.
⑫ 공개키 인증서의 유효기간을 검사한다. 유효기간 필드의 값이 현재 날짜와 같은지 큰지를 비교한다. 현재 날짜보다 작다면 유효기간 만료로 처리한다.
⑬ IC 카드 공개키 알고리즘 식별자를 인식할 수 있는지 검사한다.
⑭ 모든 검사를 성공했다면, IC 카드 EMV 인증서 검증을 완료한다.

2.4 IC 카드 인증메커니즘

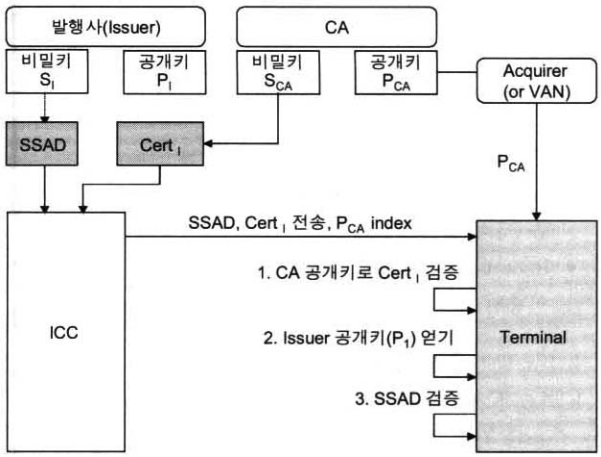
IC카드의 위변조를 방지하기 위해 EMV 규격은 SDA(Static Data Authentication)와 DDA(Dynamic Data Authentication) 2가지 방식의 IC 카드 인증메커니즘을 정의하고 있다[3]. RSA 암호를 제공하지 못하는 IC 카드에서 사용되는SDA(Static Data Authentication) 방식은 발행사가 전자서명 값을 생성하여 IC 카드를 발급하는 방식으로 사용되는데, 단말기는 거래시 IC 카드에 저장된 발행사의 전자서명 값을 검증하여 IC 카드 위조 여부를 검증한다. RSA 암호 모듈이 내장된 IC 카드에서 사용될 수 있는 DDA 인증 메커니즘은 발행사가 IC 카드용 개인키와 EMV 공개키 인증서를 발급하여 단말기와 거래시 IC 카드가 직접 전자서명 값을 매번 생성하여 단말기에 전달하고, 이 값을 검증하는 방식이다. 각 IC 카드 인증 메커니즘을 설명하면 다음과 같다.

2.4.1 SDA(Static Data Authentication)

SDA 방식은 RSA 모듈이 없는 IC 카드를 발급하는 발행사의 서명값(Signed Static Application Data ; SSAD)을 이용해 IC 카드 자체에 대한 위변조를 확인하는 메커니즘이다. 따라서 IC 카드 발급시 발행사가 생성한 전자서명 값(SSAD)을 거래시 IC 카드 인증용으로 매번 사용되는 방식이다.

(그림 3)과 같은 SDA 검증절차는 단말기가 IC 카드에 저장된 CA 공개키 식별자, 발행사 EMV 인증서, SSAD값을 읽어와서 시작한다.

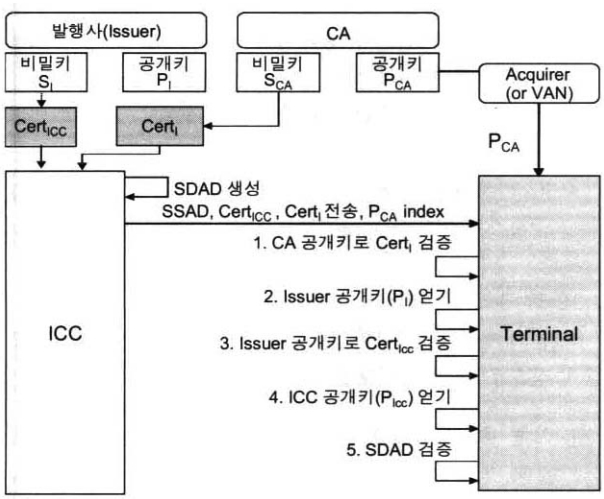
① 2.2.1 CA 공개키를 검증한다.
② 2.2.2 발행사 EMV 인증서를 검증한다.
③ SSAD 데이터의 길이가 발행사 공개키 Modulus의 길이와 같은지 비교한다. 다르다면 검증 실패로 처리한다.
④ SSAD를 복호화하여 SAD 데이터를 구한 후 Tailer가 '0xBC'와 같은지 비교한다. 다르면 복호화 실패이다.
⑤ 복호화된 데이터의 Header가 '0x6A', Signed Data Format 필드가 '0x03'과 같은지 비교한다.
⑥ 복호화된 데이터를 포함하여 해쉬 계산에 사용된 데이터를 구성한다.
⑦ 해쉬알고리즘 식별자를 사용해 해쉬값을 계산한다.
⑧ 계산한 해쉬값과 복호화한 해쉬값과 같은지 비교한다.
⑨ 위의 모든 단계가 성공적으로 끝나면 SDA는 성공이다.



(그림 3) SDA 인증 메커니즘

2.4.2 DDA(Dynamic Data Authentication)

DDA 방식은 RSA 모듈이 탑재된 IC 카드를 발급하는 발행사가 IC카드 EMV 공개키 인증서를 발급하고, 이를 이용해 IC 카드가 전자서명 값(Signed Dynamic Application Data ; SDAD)을 생성하여 IC 카드 위변조를 확인하는 메커니즘이다. 따라서 IC 카드가 생성한 전자서명 값이 거래시 IC 카드 인증용으로 매번 새롭게 생성되어 인증에 사용되는 방식이다.



(그림 4) DDA 인증 메커니즘

DDA 검증절차는 (그림 4)과 같이 단말기가 IC 카드에 저장된 CA 공개키 식별자, 발행사 EMV 인증서, IC 카드 EMV 인증서, SDAD 값을 읽어와서 시작한다.

① 2.2.1 CA 공개키를 검증한다.
② 2.2.2 발행사 EMV 인증서를 검증한다.
③ 2.2.3 IC 카드 EMV 인증서를 검증한다.
④ SDAD 데이터의 길이가 IC 카드 공개키 Modulus의 길이와 같은지 비교한다. 다르다면 검증 실패로 처리한다.
⑤ SDAD를 복호화하여 DAD 데이터를 구한 후 Tailer가 '0xBC'와 같은지 비교한다. 다르면 복호화 실패이다.
⑥ 복호화된 데이터의 Header가 '0x 6A', Signed Data Format 필드가 '0x 03'과 같은지 비교한다.
⑦ 복호화된 데이터를 포함하여 해쉬 계산에 사용된 데이터를 구성한다.
⑧ 해쉬알고리즘 식별자를 사용해 해쉬값을 계산한다.
⑨ 계산한 해쉬값과 복호화한 해쉬값과 같은지 비교한다.
⑩ 위의 모든 단계가 성공적으로 끝나면 DDA는 성공이다.

2.5 EMV PKI 분석

EMV 2000 Book 2 - Security and Key Management - 규격에 포함된 EMV PKI(Public Key Infrastructure) 분석을 통해 도출된 문제점을 요약하면 다음과 같다.

- Root CA/Local CA 공개키 배포의 어려움
  - CA 공개키 인증서 규격이 없음, 단순히 변조가 가능한 공개키 값을 저장하여 사용
  - CA 공개키 값에 대한 인증 메커니즘이 없어 변조에 취약
- 비효율적인 발행사 EMV 인증서 배포 메커니즘
  - 발급하는 모든 IC 카드마다 발행사(Issuer) EMV 인증서를 저장
  - IC 카드는 저장공간의 제약(보통 8Kbyte 32Kbyte)을 지니고 있는 상황에서 IC 카드/단말기 저장 공간 낭비 초래
- 비효율적인 IC 카드 인증메커니즘
  - SDA IC 카드 인증메커니즘의 경우 거래마다 POS 단말기에 발행사 EMV 인증서 전송
  - DDA IC 카드 인증메커니즘의 경우 거래마다 POS 단말기에 발행사 및 IC 카드 EMV 인증서를 전송
  - IC 카드 처리속도의 제약으로 인해 거래처리 속도가 매우 중요한 상황에서 SDA, DDA와 같은 IC 카드 인증메커니즘 적용시 트랜젝션 처리시간 증가 요인으로 작용, 이는 IC 카드 활성화에 직접적인 악영향을 초래

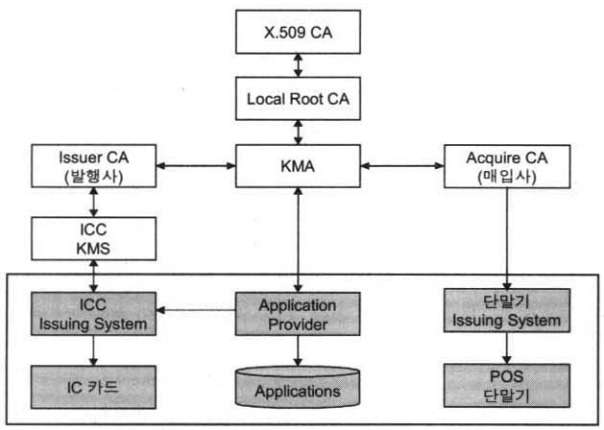
이와 같은 문제점을 해결하기 위해 본 논문에서 X.509 인증서를 도입하여 "EMV기반 Payment PKI 모델" 제안하고 CA 시스템을 설계하고, 구현하였다. 이와 함께 제안한 Payment PKI 모델을 통해 기존의 EMV IC 카드 인증메커니즘(SDA, DDA)의 운영환경 및 성능을 개선할 수 있는 "효율적인 IC 카드 인증메커니즘(SDAx, DDAx)"을 구현하는 것이 가능하도록 하였다.

3. EMV 기반 Payment PKI 모델

EMV기반 Payment PKI에 대한 구현 모델를 제시하고, 이를 이용한 "효율적인 IC 카드 인증메커니즘"을 제안하고자 한다.

3.1 EMV 기반 Payment PKI 모델

본 논문에서 제안하고 있는 EMV 기반 Payment PKI 구조는 (그림 5)과 같다. 먼저 각 구성요소의 역할을 살펴보면 다음과 같다.



(그림 5) EMV 기반 Payment PKI 구성도

3.1.1 로컬 Root CA/X.509 CA

로컬 Root CA는 자체적으로 공개키/비밀키 쌍을 생성한 후 운영에 들어가는데, 이때 CA 공개키 관리의 형태는 크게 EMV 공개키 규격, X.509 CA와 연동하여 X.509 공개키 인증서 규격을 사용하는 방식을 선택적으로 이용할 수 있도록 하였다.

Payment PKI에서 정의하고 있는 로컬(Regional) Root CA의 기능 및 역할을 요약하면 다음과 같다.

- Local(Regional) CA의 개인키/공개키 쌍을 생성한다.
- Local(Regional) CA의 개인키는 IC 카드를 이용해 안전하게 보관한다.
- X.509 CA으로부터 Local(Regional) CA의 X.509 공개키 인증서를 발행받는다.
- 발행사 CA용 EMV 공개키 인증서를 발행한다.
- CA 공개키 또는 X.509 공개키 인증서는 KMA(Key Management Authority)를 통해 발행사(Issuer)와 매입사(Acquirer)에 배포된다. 인증서 형태가 아닌 CA 공개키를 같이 배포하는 이유는 기존 규격과의 호환성을 유지하기 위해서이다.

3.1.2 KMA(Key Management Authority)

발행사(Issuer) CA EMV 공개키 인증서 발급 요청을 수신하여 Payment Root CA로부터 공개키 인증서를 발급 해주는 에이전트(Agent)이다. Payment PKI하의 모든 하위 CA가 인증서 발급 프로토콜을 통해 연결된다[8, 9].

- 발행사 EMV 공개키 인증서 발급 서비스
- Application Provider에게 IC 카드용 어플리케이션 인증 서비스
- Root CA X.509 공개키 인증서, 발행사 EMV 공개키 인증서

3.1.3 발행사(Issuer) CA

IC카드를 발급하는 발행사(Issuer)가 운영하는 CA의 역할은 다음과 같다.

- 발행사(Issuer) CA의 개인키/공개키 쌍을 생성한다.
- 로컬 Root CA로부터 발행사 EMV 공개키 인증서를 발행 받는다.
- IC 카드 EMV 공개키 인증서를 발행한다.

3.1.4 ICC KMS

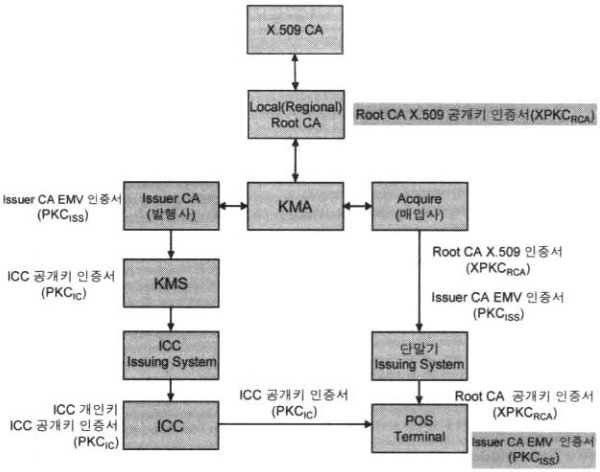
IC 카드 발급단계에서 IC 카드 성능 상 자체에서 공개키 쌍을 생성하는 것은 불가능하기 때문에 IC 카드 발급시스템(ICC Issuing System)과 연동하여 IC 카드용 키를 생성하고 관리하기 위해 필요한 구성요소이다.

- IC 카드용 개인키와 공개키를 생성한다. EMV 규격에서는 발행사의 역할로 정의하고 있다.
- 발행사(Issuer) CA로부터 IC 카드 EMV 공개키 인증서를 발행 받는다.
- IC 카드 EMV 공개키 인증서와 개인키를 IC 카드 발급시스템(ICC Issuing System)에 안전하게 전송한다.

3.2 키 관리 시나리오

(그림 6)에서 보는 바와 같이 각 구성요소 사이의 키 배포를 중심으로 한 시나리오를 살펴보면 다음과 같다.

① 로컬 CA는 X.509 공개키 인증서를 발급받아 KMA를 통해 배포한다.
② 발행사 CA는 KMA를 통해 EMV 공개키 인증서를 발급받고, 로컬 CA X.509 공개키 인증서를 가져온다.
③ 발행사 CA EMV 공개키 인증서는 KMA에서 배포되며, 이것은 매입사를 통해 단말기에 배포된다.
④ KMS는 IC 카드용 공개키를 생성하며, 발행사 CA로부터 IC 카드 EMV 공개키 인증서를 발급받아 IC 카드 발급에 사용한다.
⑤ 단말기는 매입사로부터 로컬 CA는 X.509 공개키 인증서, 발행사 EMV 공개키 인증서를 저장하고 있는다.
⑥ 거래시 단말기는 IC 카드에 저장된 IC 카드 EMV 공개키 인증서를 읽어온 후, 로컬 CA, 발행사, IC 카드 인증서의 신뢰경로를 생성하여 인증서를 검증하게 된다.



(그림 6) EMV 기반 Payment PKI 키 관리

로컬 CA는 키 관리를 위해 X.509 CA로부터 인증서를 발급 받는다. 이를 발행사와 매입사에 배포하는 방식을 통해 로컬 CA 공개키 값으로 배포되어 관리되는 기존의 방식을 보완하여 안전성을 확보하고자 하였다[7-9]. 그렇지만 단말기는 X.509 인증서를 검증할 수 있는 기능이 추가해야 하는 부분이 있다. 또한 발행사의 EMV 인증서를 IC 카드에 저장하여 배포하던 기존의 방식을 수정하여 이를 매입사를 통해 직접 단말기에 배포하여 효율성를 향상시키도록 하였다.

3.3 효율적인 IC 카드 인증메커니즘

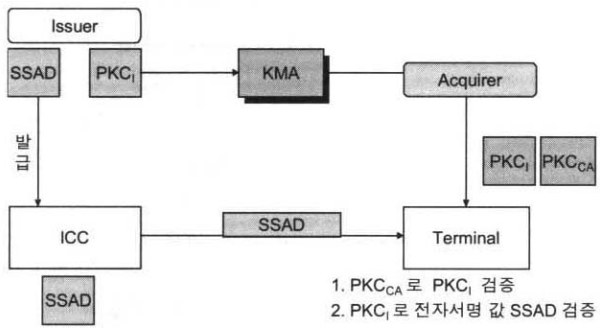
EMV 기반 Payment PKI 모델을 통해 다음과 같이 2가지 효율적인 IC 카드 인증메커니즘을 운영하는 것이 가능하다. 이와 같은 인증 메커니즘은 발행사 EMV 공개키 인증서 배포 방식을 바꾸고, IC 카드 발급정보를 감소시킴으로써 IC 카드 저장 공간을 최소화하고, IC 카드 대량 발급시 효율성을 증가시키는 것이 가능해진다. 또한 거래시 매번 전송되는 발행사 CA 인증서를 전송하지 않도록 함으로써 거래 트랜젝션 소요시간을 줄이는 것이 가능해지는 장점을 지닌다.

3.3.1 SDAx 인증절차 및 검증

SDAx IC 카드 인증메커니즘을 사용하기 위해 IC 카드에 발급되는 정보는 발행사가 전자서명한 SSAD(Signed Static Application Data)값이다. 기존의 SDA 인증메커니즘과달리 발행사 EMV 인증서는 발급 정보에 포함하지 않는다. 대신 KMA를 통해 발행사 EMV 인증서를 매입사에 전달하여 단말기에 배포해야 한다. 매입사에 배포하는 것은 온라인으로 처리하는 것이 가능하다.

SDAx 인증절차는 다음과 같다((그림 7) 참조).

① 단말기는 미리 CA는 X.509 공개키 인증서, 발행사 EMV 공개키 인증서를 저장하고 있는 상태이다. 따라서 기존의 방식과 달리 거래마다 발행사 EMV 공개키 인증서를 읽을 필요가 없다.
② 거래시 단말기는 IC 카드에 저장된 SSAD만을 읽어온다.
③ 단말기는 SSAD를 수신한 후 CA X.509 공개키 인증서와 발행사 EMV 공개키 인증서를 검색한다.
④ CA X.509 공개키 인증서를 검증한다.
⑤ CA 공개키를 이용해 발행사 EMV 공개키 인증서를 검증한다.
⑥ 발행사 CA 공개키를 이용해 발행사가 전자서명한 SSAD 값을 검증한다.



(그림 7) SDAx 인증절차

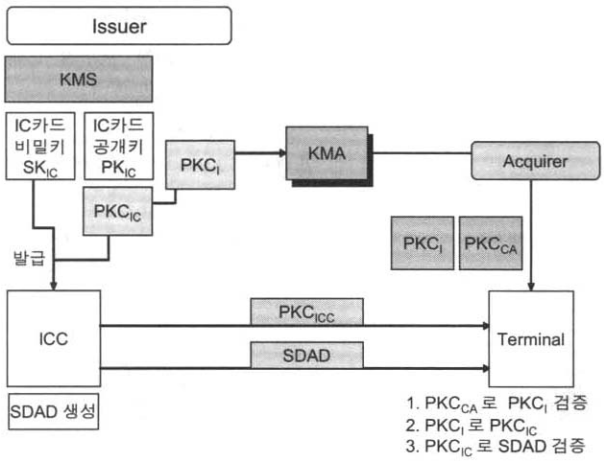
이와 같은 SDAx는 IC 카드 발급시 발행사 인증서를 발급정보에서 제외시킴으로써 IC 카드 저장 메모리를 감소시키는 것이 가능하며, 거래시 발행사 인증서를 전송하지 않기 때문에 트랜젝션 처리 시간을 감소시키는 것이 가능하다.

3.3.2 DDAx 인증절차 및 검증

DDAx IC 카드 인증메커니즘을 사용하기 위해 IC 카드에 발급되는 정보는 IC 카드 개인키와 공개키 인증서 두 정보이다. SDAx와 같이 기존의 DDA 인증메커니즘과 달리 발행사 EMV 인증서는 발급 정보에 포함하지 않는다. 대신 KMA를 통해 발행사 EMV 인증서를 매입사에 전달하여 단말기에 배포해야 한다.

DDAx 인증절차는 다음과 같다((그림 8) 참조).

① 단말기는 미리 CA는 X.509 공개키 인증서, 발행사 EMV 공개키 인증서를 저장하고 있는 상태이다. 따라서 기존의 방식과 달리 거래마다 발행사 EMV 공개키 인증서를 읽을 필요가 없다.
② 거래시 IC 카드는 개인키를 이용해 SDAD(Signed Dynamic Application Data)를 생성한다.
③ IC 카드 EMV 인증서와 함께 SDAD를 전송한다.
④ 단말기는 SDAD를 수신한 후 CA X.509 공개키 인증서와 발행사 EMV 공개키 인증서를 검색한다.
⑤ CA X.509 공개키 인증서를 검증한다.
⑥ CA 공개키를 이용해 발행사 EMV 공개키 인증서를 검증한다.
⑦ 발행사 CA 공개키를 이용해 IC 카드 EMV 인증서를 검증한다.
⑧ IC 카드 공개키를 이용해 IC 카드가 전자서명한 SDAD 값을 검증한다.



(그림 8) DDAx 인증절차

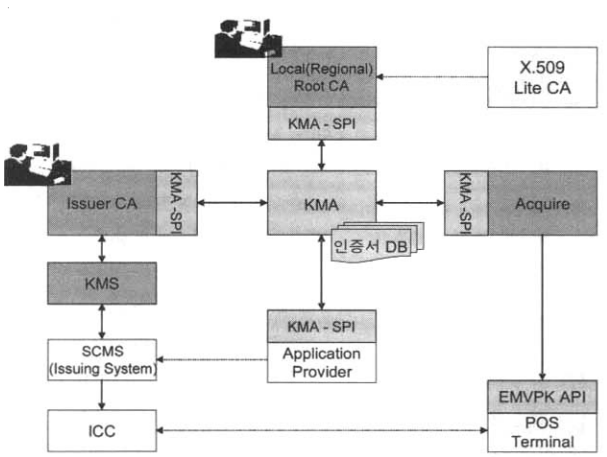
이와 같은 DDAx는 IC 카드 발급시 발행사 인증서를 발급정보에서 제외시킴으로써 IC 카드 저장 메모리를 감소시키는 것이 가능하며, 거래시 발행사 인증서를 전송하지 않기 때문에 트랜잭션 처리 시간을 감소시키는 것이 가능한 특징을 가지고 있다.

4. Payment CA 시스템 구현 및 고찰

본 장에서는 본 논문에서 제안하고 있는 Payment PKI의 구현 모델을 구현한 CA 시스템에 대해 설명을 하고, 이를 통해 구현한 "효율적인 IC 카드 인증메커니즘"을 구현하여 성능평가를 하고자 한다.

4.1 시스템 구현

본 논문에서 제안한 Payment PKI를 구축하기 위해 필요한 시스템은 (그림 9)과 같이 CA(Root/Issuer/Acquire) 시스템, KMA, KMA-SPI, KMS, EMV PK API 등과 같이 크게 5부분으로 구성되어 있다. 각 시스템의 기능을 설명하면 다음과 같다.



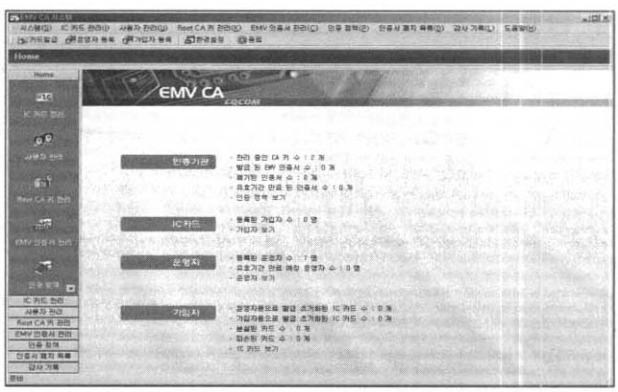
(그림 9) Payment PKI 시스템 구성도

4.1.1 CA(Root/Issuer/Acquire) 시스템

Payment PKI 구축에 있어서 핵심이 되는 CA 시스템은 EMV 공개키 인증서를 발급하는 시스템이다. 이 시스템이 제공하는 기능은 다음과 같다.

① EMV 표준 알고리즘 지원(RSA, DES/3DES, SHA-1)
② IC 카드를 이용한 키 관리(CA의 개인키와 공개키 인증서를 안전하게 관리하기 위해 IC 카드를 사용)
③ EMV 공개키 인증서 생성
④ X.509 공개키 인증서 지원
⑤ 인증서 관리 프로토콜 지원

(그림 10)는 EMV 인증서, 본 논문에서 제안한 EMV CA 시스템을 보여주고 있다. 이 시스템은 IC 카드 관리, 사용자 관리, CA 키관리, EMV 인증서 관리, 인증정책 관리 기능 등을 제공하고 있다.



(그림 10) EMV CA

4.1.2 KMA/KMA-SPI

KMA는 Root CA와 연동 인터페이스를 제공하며, 인증서 관리 프로토콜을 구현한 구성요소이다. KMA는 발행사 인증서, 매입사 인증서 등을 발급하는 서비스와 Root CA 공개키 인증서 배포 서비스, 어플리케이션 전자서명 서비스를 제공하는데, KMA-SPI는 발행사 CA와 매입사 CA에 설치되어 이와 같은 서비스를 제공한다.

4.1.3 KMS(Key Management System)

KMS는 IC 카드와 단말기용 키 생성 시스템으로써 발행사와 매입사 CA와 인증서 발급 프로토콜을 통해 연동되어 동작한다.

4.1.4 EMV PK API

본 논문에서 제안하고 있는 IC 카드 인증메커니즘을 구현하기 위해 단말기에 탑재되어 동작하는 EMV 암호 API이다. 추가적으로 X.509 인증서 검증 기능이 포함되어 있다.

이 모듈은 EMVco에서 실행하고 있는 EMV Level 2 테스트를 통해 EMV 보안 규격에 포함되는 모든 테스트 항목에 대해 테스트를 수행하여 검증을 완료한 상태이다.

4.2 IC 카드 인증메커니즘 비교분석

기존의 EMV IC 카드 인증메커니즘과 제안 IC 카드 인증메커니즘을 비교 분석한 내용을 살펴보면 다음과 같다. 본 논문에서 제안한 Payment PKI 모델을 통해 구현한 CA시스템은 인증서 관리 부분을 보완하여 효율적인 인증서 배포가 가능하다. 이를 통해 IC 카드 발급시 발급정보의 크기를 줄임으로써 최대 1K의 메모리 절약이 가능해지고, 또한 IC 카드 대량 발급시 소요 시간을 크게 줄이는 것이 가능해진다. 또한 거래 처리시 IC 카드 인증 절차를 반드시 수행해야 하는데, 이때 인증서 배포를 위한 트랜잭션을 제외시킴으로써 거래 트랜잭션 처리 시간을 줄이는 것이 가능하다.

<표 4>은 본 논문에서 제안한 "효율적인 IC 카드 인증메커니즘"의 구현을 통해 성능 평가를 수행한 결과를 보여주고 있다. 이 테스트에 사용된 IC 카드는 16비트 Schlumberger 자바카드와 잼플러스사의 IC 카드 리더 GCR410을 사용했으며, 단말기 환경은 PC(P3 866)로 구성하였다. 이와 같은 성능평가의 결과는 다음과 같다.

〈표 4〉 IC카드 인증메커니즘 성능평가

비교항목	EMV SDA	제안 SDAx	EMV DDA	제안 DDAx
CA 공개키 배포	ICC	단말기	ICC	단말기
ICC내 인증서 저장 개수	1개	0개	2개	1개
ICC 내 저장공간 크기(바이트)	167-287	0	308-460	141-173
ICC 발급시간(초)	6.910	5.608	7.450	6.148
발행사 인증서 배포	거래시 매번	1회	거래시 매번	1회
트랜잭션 (SDA/DDA)	2회	1회	3회	2회
성 능 (처리소요시간, 초)	7.112	5.909	9.190	7.985

먼저 IC 카드 발급시 저장되는 발행사 인증서를 제외시킴으로써 IC 카드 내에 저장되는 발급정보를 167(공개키 길이가 1204비트인 경우)-287(공개키 길이가 1984비트인 경우) 바이트까지 줄이는 것이 가능하다. 이를 통해 1장의 IC 카드를 발급하는데 소요되는 시간을 1.103초에서 1.302초까지 줄이는 것이 가능한 장점을 지니며, 이는 대량 발급되는 환경을 고려해볼 때 10,000장의 IC 카드를 발급하는 경우 3.06시간을 절감하는 효과를 얻을 수 있다.

또한 발행사 인증서 배포의 경우 거래 시 매번 발생하는데 비해, 단말기에 배포함으로써 그 회수를 1회로 줄임으로써 거래처리 소요 시간을 1.203초 정도를 줄이는 것이 가능하다. 따라서 IC 카드 이용의 효율성을 증대하는 장점을 지닌다.

5. 결 론

전자상거래 분야에서 공개키 기술을 안전하게 이용하기 위해서는 PKI 구축이 반드시 선행되어야 한다. 인터넷 환경을 위한 X.509 인증서 기반의 인터넷 PKI, 무선망 환경을 위한 WAP PKI 등은 이미 시스템 구축 및 서비스 단계에 있으나 상대적으로 금융망 환경에 적합한 EMV PKI에 대한 연구가 미진한 상태이다. 최근 마그네틱 형태의 금융카드가 IC카드로 교체되는 시점에서 EMV 표준을 기반으로 IC 카드와 접목된 형태로 신용카드 결제시스템이 개발되는 상황에서 금융환경에 특화된 EMV PKI의 구축의 필요성은 더욱 중요하다 할 수 있다.

본 논문에서는 EMV 규격에 포함된 EMV 인증서를 기반으로 EMV PKI 모델을 제시하고, EMV PKI 구축에 필요한 CA 시스템을 설계하고, 구현하였다. 또한 SDA와 DDA 방식의 IC 카드 인증메커니즘의 운영환경 및 성능을 개선할 수 있는 "효율적인 IC 카드 인증메커니즘(SDAx, DDAx)"을 제안하고 구현을 통해 성능평가를 하였다. 이를 통해 금융환경에 특화된 Payment PKI 모델을 제시함으로써 IC카드 기반 신용카드시스템 개발 분야에 적용 가능하며, 효율적인 IC카드 인증메커니즘을 제안하여 IC카드 메모리 낭비 감소, 거래 처리 시간 단축, 효율적인 운영환경 및 성능향상을 가능하게 할 것으로 기대된다.

참 고 문 헌

[1] 금융결제원, 금융IC카드표준 Version 1.1, Feb., 2003.

[2] EMVco, EMV 2000 BOOK 1 : Application Independent ICC to Terminal Interface Requirements Version 4.0, Dec., 2000.

[3] EMVco, EMV 2000 Book 2 : Security and Key Management Version 4.0, Dec., 2000.

[4] EMVco, EMV 2000 Book 3 : Application Specification Version 4.0, Dec., 2000.

[5] EMVco, EMV 2000 Book 4 : Cardholder, Attendant, and Acquirer Interface Requirements Version 4.0, Dec., 2000.

[6] VISA, Visa Integrated Circuit Card Specification Version 1.4.0, Mar., 2001.

[7] R. Housley, W. Ford, W. Polk, D. Solo, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF RFC 2459, Jen., 1999.

[8] C. Adams, S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," IETF RFC-2510, Mar., 1999.

[9] M. Myer, R. Ankney, A. Malpani, S. Galperin, C. Adms, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP," IETF RFC2560, June, 1999.

송 상 헌

e-mail : hern@cqcom.com
1995년 배재대학교 전자계산학과 학사
1997년 충남대학교 대학원 컴퓨터과학과 석사
2001년 충남대학교 대학원 컴퓨터과학과 박사
2000년~현재 ㈜ 시큐컴
주관심 분야 : 전자지불시스템, 네트워크 보안

최 석 진

e-mail : choisj@etri.re.kr
1995년 경북대학교 전자공학과 학사
1998년 한국과학기술원 전기및전자공학과 석사
1998년~2000년 LG 반도체 연구원
2000년~현재 국가보안기술연구소 선임연구원
주관심 분야 : 암호알고리즘, XML 보안, PKI

류 재 철

e-mail : jcryou@home.cnu.ac.kr
1985년 한양대학교 산업공학과(학사)
1988년 Iowa State Univ.(전산학 석사)
1990년 Northwestern Univ.(전산학 박사)
1991년~현재 충남대학교 정보통신공학부 교수
2003년~현재 충남대학교 인터넷 침해대응기술연구센터장
관심분야 : 인터넷 보안