저사양 스마트카드의 PQC 인증 방안에 관한 연구

류지은[1], 김덕상[2], 강주성[1], 염용진[1]*
국민대학교[1], ㈜에잇바이트[2]
{ofryuji, jskang, *salt}@kookmin.ac.kr, deoksang.kim@8byte.co.kr

A Study on the Authentication using PQC for Resource-Constrained Smart Card

Jieun Ryu[1], Deoksang Kim[2], Ju-sung Kang[1], Yongjin Yeom[1]*
Kookmin Univ.[1], 8BYTE, Inc.[2]

요 약

양자컴퓨팅 기술의 발전으로 기존 공개키 암호의 안전성이 위협받게 됨에 따라, 2016년부터 양자내성암호(post-quantum cryptography, PQC) 표준화 및 전환이 이뤄지고 있다. 스마트카드와 같은 저사양 환경도 PQC 도입이 필요하지만, 높은 연산 복잡도와 큰 입출력 데이터 크기 문제로 스마트카드에 표준 구현된 PQC를 적용하기 어려운 상황이다. 이에 본 논문은 기존 공개키 암호 기반 스마트카드 인증 절차를 분석하고, 인증 앱이 실행되는 단말의 연산 자원을 활용하여 PQC 인증을 수행하는 방안을 제안한다.

Ⅰ. 서론

스마트카드는 하나 이상의 IC 칩(integrated circuit chip)을 내장하여, 마이크로프로세서(microprocessor, MPU)와 운영체제로 연산 기능을 제공하고 EEPROM(electrically erasable programmable read-only memory)을 안전한 저장영역으로 제공하는 전자식 카드이다[1]. 초기 스마트카드는 데이터 저장 기능만을 제공했으나, 금융 및 인증 등의 서비스에 널리 활용되면서 암호화 기능을 탑재하도록 발전했다. 스마트카드는 암호화 기능을 제공함과 동시에 물리적으로 소유한 사용자만 스마트카드에 내장된 데이터에 접근 가능하다는 특성으로 높은 보안성을 제공하며, 이에 기반하여 개인정보와 밀접한 분야에서 폭넓게 활용되고 있다[2].

한편, 양자컴퓨터의 급격한 발전에 따라 기존 공개키 암호 및 서명 알고리즘의 안전성이 위협받게 되자, 이에 대응하기 위한 양자내성암호(post-quantum cryptography, PQC)가 연구되고 있다. 2022년 미국의 첫 PQC 표준 알고리즘 4종이 선정되었으며, 최근에는 양자컴퓨터를 활용한 보안 시스템 공격이 현실화되기 전, 기존 암호 시스템을 PQC로 전환하려는 노력이 이어지고 있다[3].

스마트카드 역시 스마트카드 인증이나 사용자 인증 등에 암·복호화와 전자서명을 수행하므로 암호 시스템을 PQC로 전환해야 하나, 주로 16-bit 또는 32-bit 프로세서로 개발된 현대 스마트카드는 연산량이 많고 데이터 크기가 큰 PQC를 구동하기 어렵다는 한계가 있다. 특히 스마트카드는 한 번 발급받으면 재발급 주기가 5~10년 정도로 매우 길기 때문에, 64-bit 프로세서의 고성능 스마트카드를 개발하는 것과 별개로 기존 발급된 스마트카드들에 대한 PQC 적용 연구가 필요하다.

따라서 본 논문에서는 스마트카드 기반 인증 절차를 분석하고, 저사양 환경인 스마트카드에 PQC를 도입하는 방안으로 외부 단말의 연산 능력을 활용하는 방식을 제안한다. 향후 본 연구의 제안 모델을 발전시켜 실제 스마트카드에 적용하면 전자 신분증의 인증이나 모바일 결제 시 수단 추가 인증 등에 높은 활용 가치를 가질 것으로 기대된다.

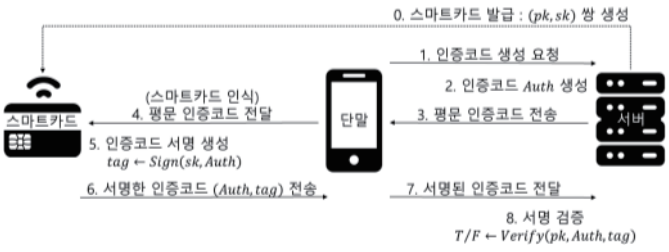
Ⅱ. 스마트카드의 인증 절차

본 논문에서는 스마트카드 인증을 인증에 사용된 카드가 서버에서 요구하는 올바른 개인키를 소유했는지 검증하는 과정으로 정의한다. 인증 과정에는 스마트카드, 이를 발급·관리하는 서버, 서버가 배포한 인증 앱(application, App)이 설치된 단말이 참여한다. 해당 과정은 서버가 각 스마트카드를 위한 개인키/공개키 쌍을 생성하고, 스마트카드 내부에 개인키를 외부 노출 없이 안전하게 저장함을 전제로 수행된다.

스마트카드 인증은 인증 후 제공되는 서비스의 특성이나 인증 환경의 특성 등을 고려하여 암호화 또는 전자서명을 통해 이뤄진다. 본 장에서는 두 방식 중 더 보편적으로 사용되는 전자서명 방식을 예로 스마트카드 인증 절차를 설명한다. <그림 1>은 해당 스마트카드 인증 절차 흐름도이다.

스마트카드 인증 절차

0. 스마트카드 인증에 앞서 카드의 발급이 선행된다. 서버가 발급하려는 스마트카드를 위한 개인키-공개키 쌍 (sk, pk)을 생성하며, 개인키 sk를 내장한 스마트카드를 발급하고 카드 고유번호 또는 카드 발급 신청자명 등의 고유 정보인 id와 함께 공개키 pk를 저장한다.
1. 단말이 서버에 인증코드 생성을 요청하여 스마트카드 인증을 시작한다. 필요에 따라 단말은 id를 함께 전달할 수 있다.
2. 서버가 단말의 요청에 따라 인증코드 $Auth$로 세션키(session key)나 OTP (one-time password) 등을 생성한다.
3. 서버가 인증코드 $Auth$를 평문(plaintext) 상태로 단말에 전송한다.
4. 단말이 스마트카드 인식 요청 메시지를 띄우고, 인식된 스마트카드에 평문 상태인 인증코드를 전달한다.
5. 스마트카드가 올바른 개인키를 내장한 경우, 이를 사용하여 평문 상태인 인증코드에 $tag = Sign(sk, Auth)$로 서명하여 서명된 인증코드 $(Auth, tag)$를 생성한다.
6. 스마트카드가 서명한 인증코드를 단말에 전송한다.
7. 단말이 스마트카드로부터 수신한 $(Auth, tag)$를 서버에 전달한다.
8. 서버가 sk에 대응되는 pk를 이용해 $Verify(pk, Auth, tag)$를 계산하고 서명이 올바른지 확인함으로써 스마트카드를 검증한다.



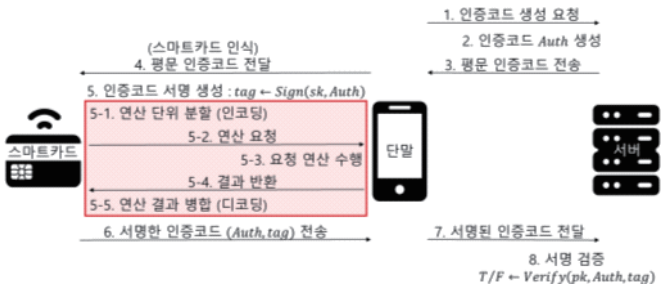
<그림 1> 스마트카드의 인증 절차

암호화 기반 인증 방식도 이와 비슷한 절차로 수행되지만, 서버가 암호화된 인증코드를 전송하면 스마트카드가 서명이 아닌 복호화 연산을 수행한다는 차이가 있다.

Ⅲ. PQC 적용 방안

현재 스마트카드 인증에 사용되는 암호 알고리즘은 PQC와 비교하여 연산량이 적고 저사양 환경을 위한 전용 하드웨어 가속기가 개발되어 있으므로 스마트카드 내에서 모든 연산을 처리할 수 있다. 그러나 PQC는 기존 공개키 암호 알고리즘과 다른 수학적 문제에 기반하므로 새로운 구성 함수를 사용하기 때문에 기존 가속기를 활용할 수 없다. 또한, PQC의 각 함수는 연산 입출력 데이터 크기가 매우 크기 때문에 연산량이 많아 현실적인 시간 내 연산 불가능하다는 문제가 있다.

이러한 문제를 해결하기 위해, 스마트카드에서 이뤄지는 PQC 연산의 일부를 외부 단말의 연산 능력을 활용하여 계산하는 방식을 제안한다. 단, 이러한 분산 연산 과정을 악용하려는 단말이 추후 인증 과정에서 스마트카드 없이 인증에 성공할 수 없도록 스마트카드에 내장된 개인키는 추가적인 인증 없이 단말에 노출되지 않게 보호되어야 한다. 제안하는 모델에서는 개인키의 노출을 방지하기 위하여 분산 연산의 입출력에 인코딩(encoding)을 적용한다. <그림 2>의 붉게 표시된 영역은 스마트카드 인증 절차 중 연산의 일부가 단말에서 수행되는 부분을 나타낸다.



<그림 2> 단말의 연산 능력을 활용한 스마트카드 인증 절차

이러한 방식은 안전성과 효율을 고려하여 스마트카드 내에서 연산하기 어려운 고비용 연산에만 적용한다. 또한, 개인키를 포함한 함수의 입력 데이터에 랜덤(random) 인코딩을 적용하여 개인키를 구별할 수 없게 만듦으로써, 단말이 개인키를 복구할 수 없도록 한다.

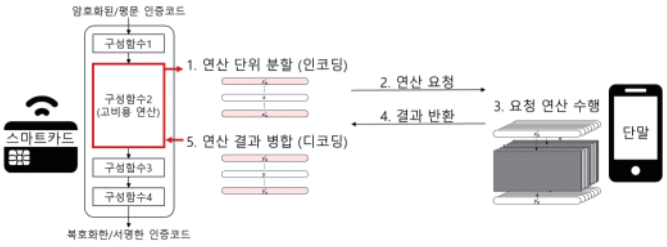
$$y = f(x) \longleftrightarrow \tilde{y} = \tilde{f}(\tilde{x})$$

이때, $\tilde{x}$는 인코딩된 x, $\tilde{y}$는 인코딩된 y이며, x에 대한 인코딩과 $\tilde{y}$에 대한 디코딩 과정은 스마트카드에서 이뤄진다. $y = f(x)$는 스마트카드에서 이뤄지는 연산이고 $\tilde{y} = \tilde{f}(\tilde{x})$는 단말에서 이뤄지는 연산이다.

2장에서 설명한 전자서명을 이용한 인증 절차 중 스마트카드에서 수행하기 어려운 고비용 연산을 g라고 하자. 스마트카드는 g의 입력이 되는 중간 값 x와 개인키 sk, g의 출력인 y에 대하여 다음과 같이 단말로의 연산 요청 식을 표현할 수 있다.

$$y = g(sk, z) \longleftrightarrow \tilde{y} = \tilde{g}(\tilde{sk}, \tilde{x})$$

인코딩 방식의 예로 <그림 3>과 같이 스마트카드에 사전 저장된 랜덤 데이터를 이용하여 추가 입력을 생성하는 방식이 있다. 추가 입력으로 인코딩을 적용하고 단말에 실제 데이터를 포함하여 모든 데이터에 대한 연산을 요청하면 단말은 실제 데이터가 무엇인지 구별할 수 없다. 반면 스마트카드는 반환 결과 중 실제 데이터에 대한 결과만을 걸러내는 디코딩(decoding)을 통해 개인키 노출 없이 함수의 결과를 얻을 수 있다.



<그림 3> 추가 랜덤 입력 데이터를 사용한 단말의 연산 능력 활용 방식

또 다른 방식으로는 <그림 4>와 같이 데이터의 연산 단위를 분할하고, 이에 대하여 스마트카드 내에서 연산 가능한 작은 비선형 인코딩을 적용하는 방식이 있다. 단말에 각 블록 단위로 함수 연산을 요청하고 결과를 받으면, 스마트카드는 이를 디코딩한 뒤 병합하여 전체 연산 결과를 얻을 수 있다. 상황에 따라 두 방식을 혼용할 수 있으며, 이 외에도 개인키를 보호하기 위해 다양한 인코딩 방식을 적용할 수 있다.

<그림 4> 비선형 랜덤 인코딩를 사용한 단말의 연산 능력 활용 방식

현재 현장에서 사용되는 스마트카드의 프로세서가 16-bit 또는 32-bit임을 고려할 때, 표준 PQC 중 암·복호화 기능을 제공하는 KEM(key encapsulation mechanism)을 사용하면 위 방식을 적용할 수 있을 것으로 기대된다. 표준 PQC 전자서명은 KEM보다 키 크기, 통신 데이터 크기, 기준 자료형 등이 매우 크고 요구하는 연산 능력 높다. 그러나 표준 KEM은 저사양 32-bit 환경에 구현 가능하며, 비교적 쉽게 16-bit 단위 알고리즘으로 수정할 수 있다[4].

GP(GlobalPlatform) 2.1.1을 따르는 Java card 2.2.0 환경에서 실험한 결과, 16-bit 프로세서 기반 스마트카드에 KEM 표준인 ML-KEM을 적용하면 1회 스마트카드 인증에 45초 이상이 소요되어 상용 서비스에는 부적합함을 확인했다. 그러나 제안한 방식을 적용하면 단말 측 연산은 매우 빠른 속도로 수행되어 그 소요 시간을 무시할 수 있으므로, 스마트카드와 단말 간의 통신 소요 시간만을 고려할 경우 최대 5초 이하까지 성능 향상이 가능할 것으로 기대된다.

Ⅳ. 결론

본 논문에서는 스마트카드와 같은 저사양 환경에서 외부 단말의 연산 능력을 활용하여 PQC 인증 연산을 수행하는 방법을 제안하였다. 제안된 방식은 이미 발급된 스마트카드에도 PQC를 적용할 수 있도록 하여, 스마트카드의 재발급 없이 펌웨어 업데이트만으로 인증 안전성을 향상시킬 것으로 예상한다. 향후 연구에서는 스마트카드 환경에 표준 KEM을 이용한 인증 모델을 구체화하고, 설계·제안한 인증 모델의 안전성을 분석한다.

ACKNOWLEDGMENT

이 논문은 서울시 산학연 협력사업 2025년도 양자 기술개발 지원사업 (QR250002, 양자내성암호 Kyber를 이용한 스마트카드 인증기술 개발)의 지원을 받아 수행된 연구임

참 고 문 헌

[1] Rankl W. and Effing W., “Smart card handbook,” John Wiley & Sons, pp. 1-25, 2004.

[2] Taherdoost H., Sahibuddin S. and Jalaliyoon N., “Smart card security: technology and adoption,” International Journal of Security, 5(2), 74-84, 2011.

[3] Barker W., Polk W. and Souppaya M., “Getting Ready for Post-Quantum Cryptography: Exploring Challenges Associated with Adopting and Using Post-Quantum Cryptographic Algorithms,” NIST Cybersecurity White Paper(CSWP) 15, 2021, (https://doi.org/10.6028/NIST.CSWP.15)

[4] Moody D., Perlner R., Regenscheid A., Robinson A. and Cooper D., “Transition to Post-Quantum Cryptography Standards,” NIST IR 8547 ipd. https://doi.org/10.6028/NIST.IR.8547.ipd