1:N 암호통신프로그램 키 설정 과정의 Kyber 적용 가능성에 관한 연구

원희정[1], 최 찬[1], 류지은[2], 강주성[1,2], 염용진[1,2]*
국민대학교 정보보안암호수학과[1] / 금융정보보안학과[2]
{hiijng1220, chanchan_918, ofryuji, jskang, *salt}@kookmin.ac.kr

A Study on the Applicability of Kyber in the Key Setup Process based on the 1:N Quantum Cryptographic Communication Programs

Heejeong Won[1], Chan Choi[1], Jieun Ryu[2], Ju-Sung Kang[1,2], Yongjin Yeom[1,2]*
Dept. of Information Security, Cryptology, and Mathematics[1]/
Financial information security[2], Kookmin Univ.

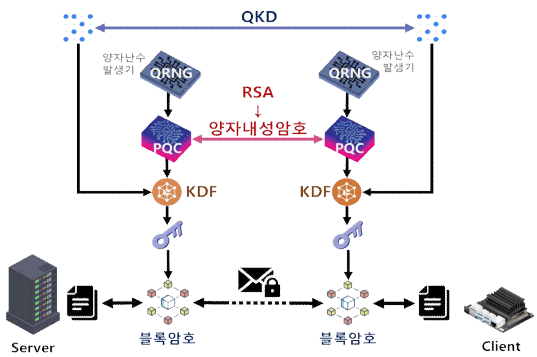
요 약

2018년 제안된 1:N 암호통신프로그램은 양자키분배(QKD) 기술과 RSA를 결합한 이중키 설정 키유도함수(KDF)를 사용하여 키 생성 측면에서 효율성에 한계가 있는 QKD의 단점을 보완하면서 양자 컴퓨터 환경에서도 안전하게 키를 공유하는 통신 프로그램이다. 그러나 양자 컴퓨터 기술이 발전함에 따라 현대 공개키암호 RSA의 기반문제를 다항시간 내에 풀 수 있을 것으로 예상되므로 KDF의 RSA를 양자내성암호로 전환할 필요가 있다. 본 논문에서는 1:N 암호통신프로그램의 키 설정 과정에서 RSA를 양자내성암호 Kyber로 전환하는 과정을 설명한다. 또한 실제 1:N 암호통신프로그램의 RSA를 Kyber로 전환하고 동작을 확인함으로써, 양자내성암호 알고리즘 추가 및 파라미터 변경만으로 프로그램 흐름의 수정 없이 구현 가능함을 보인다.

Ⅰ. 서 론

양자 컴퓨터 발전에 따른 보안 분야 위협에 대비하여 암호학 분야에서는 양자 컴퓨터 환경에서도 안전한 암호를 만들기 위한 다양한 연구가 이루어지고 있다. 비밀키의 기밀성은 암호 알고리즘의 안전성에 영향을 미치기 때문에 안전한 키 공유 알고리즘을 사용하는 것이 중요하다. 이에 따라 양자 컴퓨팅 환경에서도 안전한 높은 엔트로피의 키를 생성하는 양자키분배(Quantum Key Distribution, 이하 QKD) 기술이 등장하였다. 그러나 QKD는 키 생성 측면에서 효율이 낮다는 한계가 있다. 이를 보완하기 위해 QKD와 공개키암호 기술을 결합하여 키를 생성하는 키유도함수(Key Derivation Function, 이하 KDF)에 대한 연구가 진행되고 있다[1]. 이러한 방식의 KDF로 비밀키를 생성하면 QKD를 단독으로 사용할 때보다 키 생성 효율이 높아지고, 입력한 QKD 키와 공개키암호 키 중 하나의 안전성이 위협받아도 비밀키의 안전성은 여전히 보장된다[2].

한편, 양자 컴퓨터를 사용하면 현대 주요 공개키암호의 기반문제를 다항시간 내에 풀 수 있을 것으로 예상되므로, 양자 컴퓨터 환경에 대한 현대 공개키암호의 안전성을 보장할 수 없다. 따라서 QKD와 공개키암호를 이중키로 사용하여 생성한 비밀키로 암호화한 데이터의 안전성이 QKD 안전성에 의해 보장받는 동안 KDF의 공개키암호를 양자내성암호로 전환할 필요가 있다. 본 논문에서는 [그림 1]과 같이 QKD와 RSA를 결합한 KDF로 키를 설정한 후 데이터를 암복호화하는 통신 프로그램인 1:N 암호통신프로그램에서 RSA를 Kyber로 전환하고 이후 성능 차이를 확인한다.



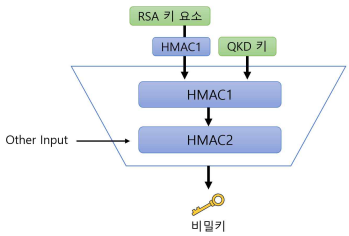
[그림 1] 1:N 암호통신프로그램의 키 생성 및 데이터 암복호화 통신 절차

Ⅱ. 1:N 암호통신프로그램

1:N 암호통신프로그램은 서버와 클라이언트가 키 설정 과정을 통해 안전하게 공유한 비밀키로 암호통신을 수행하는 프로그램이다[2]. 키 설정 알고리즘은 사용자 인증, QKD 키 인증, 키 공유 과정으로 이루어지고, 마지막 단계인 키 공유 과정에서 RSA 기반 이중키 설정 KDF를 사용하여 비밀키를 생성한다.

ⅰ. RSA 기반 이중키 설정 KDF[1]

이중키 설정은 양자 암호시스템과 공개키 암호시스템을 조합하여 고안된 방식으로, RSA를 통해 추출한 키와 QKD를 통해 추출한 키를 이중키로 사용하는 KDF의 구조는 [그림 2]와 같다.



[그림 2] RSA 기반 이중키 설정 KDF

RSA 기반 이중키 설정 KDF는 NIST 표준에서 제시한 키유도함수 모델(HMAC KDF)에 근거하여 설계되었고, HMAC KDF는 표준 해시함수 SHA-256을 내부 함수로 사용하는 HMAC으로 구성된다[3]. KDF의 동작 과정은 현대 암호 채널에서 RSA로 생성한 키와 양자 채널을 통해 공유한 QKD 키를 입력받아 HMAC1 함수를 실행하고, 해당 출력과 보안 매개변수 Other input을 확장 함수 HMAC2에 입력하여 비밀키를 생성한다. 생성한 비밀키를 통해 1:N 암호통신프로그램에서 블록암호로 암호통신을 수행한다.

ⅱ. 양자내성암호 Kyber

양자내성암호(Post-Quantum Cryptography, 이하 PQC)는 양자 컴퓨팅 환경에 대응하기 위해 새롭게 고안된 공개키암호 시스템이다. 2017년 NIST는 PQC 표준화 공모사업을 진행하였고, 2022년 최종 표준화 대상으로 선정된 격자(lattice) 기반 KEM(Key Encapsulation Mechanism) 알고리즘이 Crystals-Kyber(이하 Kyber)이다[4]. Kyber는 공개키, 개인

키 쌍을 생성하는 키 생성 알고리즘과 비밀키를 공유하는 캡슐화 알고리즘으로 구성된다. 캡슐화 알고리즘은 메시지 대신 공유할 비밀키를 생성하고, 공개키암호(Public Key Encryption, PKE)의 암호화 알고리즘을 호출하여 비밀키를 암호화한다. 이와 짝을 이루는 캡슐화 해제 알고리즘(decapsulation)은 PKE의 복호화 알고리즘을 호출하여 암호화된 비밀키를 복호화한다. Kyber를 사용하여 공유한 비밀키는 암호통신에서 메시지를 암호화하는 대칭키암호 알고리즘의 비밀키로 사용된다.

Ⅲ. 키 설정 과정의 Kyber 적용 과정

ⅰ. KDF의 입력 파라미터 차이

기존 키 공유 알고리즘에 Kyber를 적용할 때 KDF의 입력 파라미터 차이를 고려해야 한다. RSA로 공유한 값의 크기는 256bytes이고, Kyber로 공유한 값의 크기는 32bytes이므로, 공개키암호로 공유한 값의 크기 차이가 발생한다.

KDF의 입력 파라미터에 대한 크기 제약이 있을 경우, 기존과 크기가 다른 값을 입력하기 위해 내부 함수를 수정 또는 변경해야 하는 문제가 발생할 수 있다. 그러나 기존 키 공유 알고리즘은 가변길이의 입력에 대해 고정길이 32바이트를 출력하는 SHA-256를 내부 함수로 사용하는 HMAC1을 통해 이중키를 설정하므로, 입력값의 크기를 고려하지 않고 교체 가능하다[3].

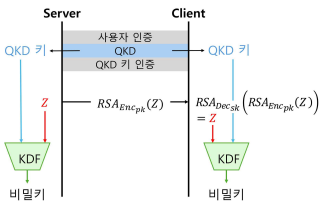
키 공유 알고리즘으로 최종 생성되는 비밀키는 HMAC1에 RSA를 입력하여 생성한 해시값과 QKD 키를 KDF에 이중키로 입력한 결과이다. 공개키암호로 생성한 값을 고정 길이로 해싱한 후에 KDF에 입력하므로, 공개키암호 RSA를 Kyber로 전환하여도 KDF는 입력 파라미터의 크기 변화에 영향을 받지 않는다.

ⅱ. 내부 암호 알고리즘 변화로 인한 키 설정 과정의 흐름 차이

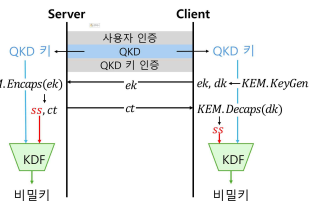
1:N 암호통신프로그램은 사용자 인증 후 QKD 키로 라운드키를 생성하고, 첫 번째 라운드키로 암호화한 난수를 공유하여 통신마다 한번씩 각 채널에 대한 인증을 진행한다. 이후 비밀키를 업데이트할 때는 별도의 인증 없이 다음 라운드키를 QKD 키로 사용하여 이중키 설정을 수행한다.

본 연구에서는 기존 키 공유 알고리즘에 Kyber를 적용할 때, QKD 키 인증 단계의 마지막에 클라이언트가 공개키와 개인키를 생성하여 서버에게 공개키를 전달하도록 구성하였다. 이는 비밀키를 업데이트할 때마다 공개키가 전송되는 것을 방지하여 통신량이 낭비되지 않도록 하기 위함이다. 또한, 기존 프로그램은 RSA를 사용하기 위해 서버와 클라이언트가 사전에 공개키와 개인키를 나눠가진 후, 통신이 시작되면 해당 키를 읽어 사용하도록 구현되어 있다. 이를 위해 서버는 클라이언트 ID마다 각기 다른 공개키를 저장해야 한다. Kyber의 경우 알고리즘 내부에서 공개키를 생성하여 전달하므로 서버는 다수의 클라이언트에 대한 공개키를 저장할 필요가 없어져 키 관리의 부담이 줄어든다.

Kyber의 키 생성 알고리즘을 제외한 두 알고리즘의 역할이 비밀키를 메시지로 받는 RSA의 암복호화 알고리즘의 역할과 동일하므로, 서버의 키 업데이트 요청에 따른 RSA와 Kyber의 통신 흐름 역시 동일하다. 이는 RSA를 Kyber로 전환할 때 키 공유 과정의 통신 절차의 수정 없이 키 설정을 수행할 수 있음을 의미한다.



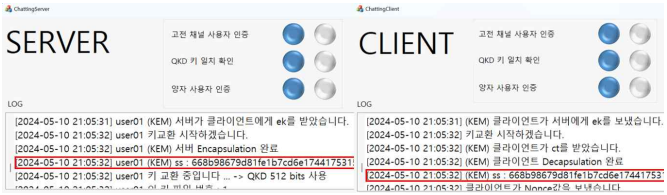
[그림 3] 기존의 RSA 기반 키 설정 과정



[그림 4] Kyber를 적용한 키 설정 과정

ⅲ. 구현 결과

1:N 암호통신프로그램에서 RSA를 Kyber로 대체하는 과정은 다음과 같다. 우선 KDF 입력값은 HMAC을 통해 해싱된 결과를 사용하므로, RSA로 공유한 키와 Kyber로 공유한 키의 크기 차이를 고려하여 KDF 함수의 입력 배열에 할당한 크기를 조정하였다. 이후 RSA를 호출하는 함수에서 encryption과 decryption을 각각 Kyber의 encapsulation과 decapsulation으로 수정하였다. 구현 결과 프로그램이 QKD 키와 Kyber 키를 결합한 비밀키를 정상적으로 공유함을 확인하였다.



[그림 5] Kyber를 적용한 1:N 암호통신프로그램 비밀키 공유 실행 화면

키 공유 알고리즘을 수정함에 따라 비밀키를 주고받는 과정에서 전송하는 데이터가 달라지기 때문에 통신량에 차이가 발생한다. 서버에서 클라이언트로 RSA 암호문 패킷을 전송할 때보다 Kyber 암호문 패킷을 전송할 때 1056bytes의 통신량이 증가한다. 또한, 클라이언트에서 서버로의 공개키 패킷 전송 과정이 통신 개설 과정에 추가됨에 따라 1614bytes의 통신량이 발생한다. 즉, 1:N 암호통신프로그램에 Kyber를 적용함으로써 발생하는 통신량 증가는 2670bytes이다. [표 1]은 RSA와 Kyber의 전송 패킷의 크기를 측정한 결과이다.

패킷크기 (header+data)	RSA	Kyber
공개키전송패킷	-	1614 (44+1570)
암호문전송패킷	558 (44+514)	1614 (44+1570)
총합	558	3228

[표 1] 암호 알고리즘에 따른 전송 패킷 크기 (단위 : byte)

Ⅳ. 결론

본 논문에서는 1:N 암호통신프로그램의 키 설정 과정에 Kyber를 적용하는 방안을 제시하고, 해당 방안을 구현하여 현대 공개키암호를 사용하는 기존 시스템에 Kyber를 적용하는 과정에서 약 2670bytes의 통신량 추가만으로 Kyber가 기존 시스템에 호환 가능함을 보였다. 이는 NIST 표준을 따르는 KDF가 가변길이 입력을 허용하고, Kyber가 RSA와 통신 흐름의 차이가 없다는 점에 기반한다. 따라서 1:N 암호통신프로그램 외에도 현대암호 기반 KDF를 사용하는 암호통신프로그램에 본 연구에서 제시한 방안과 같이 양자내성암호를 적용할 수 있을 것으로 예상한다.

ACKNOWLEDGMENT

본 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No.2021-0-00046, 국가공공 정보시스템 안전성 및 활용성 제고를 위한 차세대 암호체계 개발)과 2023년 정부(과학기술정보통신부)의 재원으로 한국연구재단-기후변화대응기술개발 사업의 지원(NRF-2021M1A2A2043893)을 받아 수행된 연구임

참 고 문 헌

[1] 박호중, 배민영, 강주성, 염용진. (2016). 양자키분배와 RSA 암호를 활용한 이중키 설정 키유도함수. 한국통신학회논문지, 41(4), 479-488.

[2] 김하은, 박호중, 강주성, 염용진. (2018). 일대다(1:N) 양자키분배 시스템 기반의 그룹통신 연구. 2018년도 한국통신학회 동계종합학술발표회 논문집.

[3] NIST, Recommendation for Key Derivation through Extraction-then-expansion, SP 800-56C, Nov. 2011

[4] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Peralta, Ray Perlner, Angela Robinson, Daniel Smith-Tone, "Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process", NIST, February 2020.