정 보 보 호 학 회 지
제 30 권 제3호, 2020. 6

격자기반 양자내성 키 교환 알고리즘 구현

박 찬 희*, 윤 영 여**, 박 해 룡***, 최 은 영***, 김 호 원**

요 약

기술의 발전으로 이론으로만 존재하던 양자 알고리즘의 실현 가능성이 보이면서 기존 암호체계의 위협이 발생할 것으로 예측하고 있다. 미국 국립표준기술연구소에서는 양자컴퓨터 환경에서도 안전한 새로운 공개키 기반의 암호체계가 필요하다는 것을 강조하였고 양자내성암호 표준을 위한 공모전을 개최하였다. 암호는 네트워크 통신에서 중요한 역할을 하고 있고 네트워크 보안 프로토콜로 TLS가 많이 사용된다. 본 논문에서는 IoT와 같은 경량 환경에 적합한 TLS 라이브러리인 mbedTLS 상에서의 격자기반 양자내성 키 교환 알고리즘을 구현하고 성능을 평가하여 양자내성암호의 사용 가능성에 대해 제시한다.

Ⅰ. 서 론

최근 슈퍼컴퓨터의 성능을 뛰어넘는 양자컴퓨터가 개발되는 등 양자컴퓨터가 계속해서 발전하고 있다[1]. 이로 인해 이론으로만 연구되었던 양자 알고리즘의 위협이 실제 발생할 것으로 예측하고 있다. 기존 암호체계는 데이터 암호화를 위한 대칭키 알고리즘과 서명, 키 교환을 위한 공개키 알고리즘으로 나눌 수 있다. 이 중 RSA, ECC와 같은 공개키 암호는 소인수분해, 이산대수문제와 같은 수학적 어려움에 기반하여 안전도를 보장하고 있다. 공개키에 사용되는 수학적 문제는 양자컴퓨터 환경에서 Shor[2] 알고리즘 같은 양자 알고리즘으로 쉽게 풀린다는 것이 증명되었고 양자컴퓨터의 발전으로 이론이 실현될 경우 기존 공개키 기반 암호체계는 안전도를 보장하지 못하게 된다. 미국 국립표준기술연구소(NIST)에서는 새로운 공개키 기반 양자내성암호 알고리즘 표준화를 위해서 공모전을 진행하고 있다.

많은 IT 분야에서 암호가 사용되고 있고 그 중에서 네트워크 통신에서 중요하게 사용된다. 현재 보안 통신을 위한 프로토콜로 TLS(Transport Layer Security)가 많이 사용되고 대표적으로 인터넷에서 거의 필수적으로 사용되는 HTTPS에서 활용되고 있다. 공개키 기반 암호의 활용도가 높은 만큼 양자컴퓨터 상용화에 따른 위협도 클 것이다. 이에 따라 TLS와 같은 보안 프로토콜에 양자내성암호를 적용하기 위한 다양한 연구가 진행되고 있다. 대표적으로 TLS 라이브러리인 OpenSSL과 SSH(Secure SHell) 라이브러리 OpenSSH에 양자내성암호를 적용하기 위한 Open Quantum Safe[3] 프로젝트가 진행되고 있다. 해당 프로젝트는 오픈소스로 진행되고 있고 몇 가지 양자내성암호 알고리즘에 대한 구현이 완료되었고 다양한 운영체제에서 동작함을 보였다. 또한, 구글에서 격자기반 양자내성암호 알고리즘인 NewHope를 TLS 라이브러리 BoringSSL에 적용하여 크롬 브라우저에서 지원하고 있다.

양자내성암호 기반의 키 교환 알고리즘의 일부는 실제 사용할 경우 16KB 이상의 파라미터 데이터를 전송해야 한다. 16KB 이상의 데이터를 TLS로 전송하기 위해 Handshake Message Fragmentation 기능을 지원해야 한다. 하지만 본 논문의 실험에서 사용할 TLS 라이브러리인 mbedTLS에서는 지원하지 않는다. 본 논문에서는 전송 데이터가 큰 격자기반 양자내성 키 교환 알고리즘을 사용하기 위한 Handshake Message Fragmentation 구현 방법을 제안하고 구현한 양자내성암호 기반 알고리즘과 기존 암호 알고리즘을 비교한 결

이 논문은 2019년 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2017-0-00616, 격자기반 양자내성 공개키암호 스킴 개발)

* ㈜스마트엠투엠 (연구원, chanhuipark@smartm2m.co.kr)
** 부산대학교 ({학생, yeo8006, 교수, howonkim }@pusan.ac.kr)
*** 한국인터넷진흥원 (연구원, { hrpark, bluecey }@kisa.or.kr)

과를 제시한다.

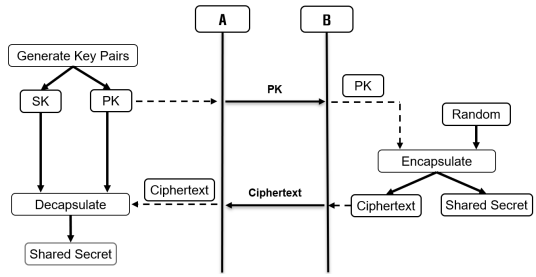
본 논문은 2장에서 격자기반 양자내성암호와 TLS에 관련된 배경지식, 3장에서는 Handshake Message Fragmentation과 같은 구현 결과를 제시하고. 4장에서는 구현한 결과물을 바탕으로 양자내성암호와 기존 암호를 비교하여 성능을 평가한다. 마지막으로 5장에서 결론을 내린다.

Ⅱ. 배경지식

양자내성암호는 안전도를 보장하는 수학적 문제 기반에 따라 격자기반(Lattice-based), 코드기반(Code-based), 아이소제니기반(Isogeny-based), 다변수기반(Multivariate-based) 등으로 나눌 수 있다. 또한, 기능에 따라 서명, 암호화, 키 설정으로 구분이 되고 본 논문에서는 키 설정에 대한 내용을 다룬다. NIST에서 주관하는 양자내성암호 표준화 공모전은 PQCrypto2016을 통해 시작하여 현재 2차 후보군 26종이 채택되었다. 격자기반 양자내성암호가 12건으로 가장 많은 비중을 차지하고 있으며 코드기반 양자내성암호 7건, 다변수기반 양자내성암호 4건, 그리고 해시기반, 아이소제니기반, 영지식 증명 기반의 양자내성암호가 각각 1건씩 선정되었다.

2.1. 격자기반 양자내성암호

격자기반 양자내성암호는 격자 상에서 SVP (Shortest Vector Problem)과 CVP(Closest Vector Problem)의 수학적 기반 어려움에 기반하고 있으며, 1996년 Ajtai[4]가 처음 제안하였다. 격자 기반 암호는 오랜 기간 연구되어 이론적으로 안전성에 대해 많이 증명되었다. 또한, 모든 연산을 행렬 곱셈과 덧셈으로 구현할 수 있어 다양한 종류 암호를 설계하기 용이하여 사물인터넷, 인공지능 등 다양한 응용 환경에서 지원이 가능하다. 본 논문에서는 격자 기반 양자내성암호 Lizard[5]를 구현하였다. Lizard는 LWE (Learning With Errors)[6]와 LWR (Learning With Rounding)[7] 문제에 기반을 둔 격자기반 알고리즘으로 잡음을 추가하여 답을 찾기 어렵게 만들어 양자컴퓨터 환경에서도 강인함을 증명하였다.



〔그림 1〕 KEM 알고리즘 동작 과정

Lizard 알고리즘은 키 교환을 위해 KEM(Key Encapsulation Mechanism)[8] 방식을 사용한다. [그림 1]을 보면 A는 공개키(PK), 비밀키(SK)를 생성하고 공개키를 전송한다. B는 공개키를 사용해서 임의의 값을 인캡슐레이트(Encapsulate)하여 암호문(Ciphertext)와 공유키(Shared Secret)을 만들고 암호문을 전송한다. A는 암호문을 받아 키와 함께 디캡슐레이트(Decapsulate)를 하면 B와 동일한 키를 공유하게 된다. 인캡슐레이트/디캡슐레이트 과정에서 양자내성암호가 사용된다.

Lizard는 KEM을 사용하는 Lizard.KEM, RLizard.KEM(Ring-Lizard.KEM)을 제안한다. 제안하는 KEM 알고리즘은 IND-CCA2 (INDistinguishability under

〔표 1〕 Lizard 알고리즘의 파라미터별 키 크기

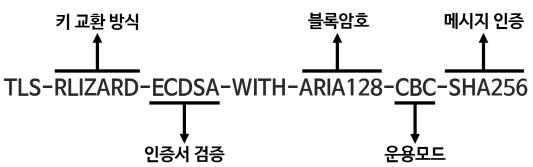
(단위 : byte)

알고리즘	파라미터	공유키	암호문	공개키	비밀키
Lizard.KEM	Category1_N536	32	17,696	1,130,496	8,608
	Category1_N663	32	10,896	1,390,592	10,640
	Category3_N816	48	26,928	1,720,320	19,632
	Category3_N952	48	31,280	1,998,848	22,896
	Category5_N1088	64	35,904	4,587,520	34,880
	Category5_N1300	64	42,688	2,727,936	41,664
RLizard.KEM	Category1	32	2,080	4,096	385
	Category3_N1024	48	4,144	4,096	641
	Category3_N2048	48	8,240	8,192	625
	Category5	64	8,256	8,192	769

adaptive Chosen Ciphertext Attack)에 대한 안전성을 제공한다. [표 1]은 제안하는 Lizard의 KEM 알고리즘에 대해 안전도를 나타내는 Category와 차원의 크기 N에 따른 파라미터를 나타낸다. Category 1/3/5는 각각 AES 128/192/256의 안전도를 나타낸다. Lizard.KEM의 경우 RLizard.KEM보다 공개키와 암호문의 크기가 크다.

2.2. TLS

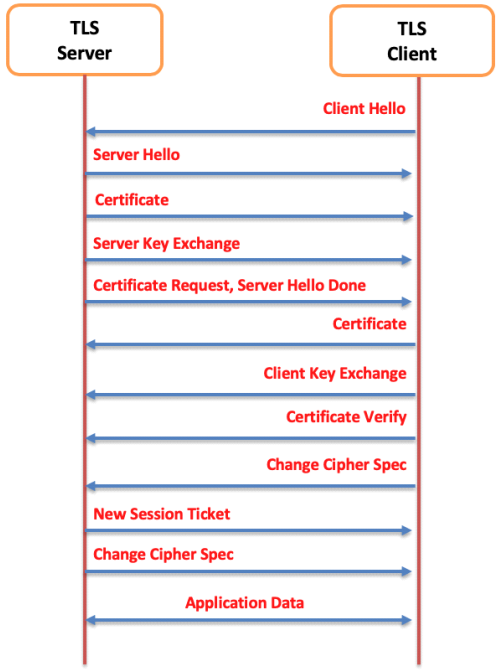
네트워크 통신의 보안을 위해 TLS 프로토콜이 사용되고 있다. TLS는 단대단 데이터 보안과 무결성을 보장하고 웹브라우저, 전자상거래, 공인인증서 등 다양한 응용 프로그램에서 사용되고 있다. TLS 표준은 통신에 사용되는 암호 알고리즘의 모음을 Ciphersuite으로 정의한다. [그림 2]와 같이 Ciphersuite은 프로토콜의 종류, 키 교환 알고리즘, 인증서 검증을 위한 서명 알고리즘, 데이터 암/복호화를 위한 블록암호, 운영모드, 메시지 인증을 위한 해시함수로 구성되어 있다. Ciphersuite에서 키 교환 알고리즘에 양자내성 알고리즘을 적용한다. TLS에서 보안 통신을 하기 위해 설정하는 단계를 Handshake라고 한다. 이 단계에서 Ciphersuite을 결정하고 통신자간의 키를 공유한다. TLS Handshake는 [그림 3]과 같은 과정을 통해 정보를 공유한다. 이 과정 중 Server/Client Key Exchange에서 양자내성암호를 적용하여 키를 교환하도록 구현한다. 앞서 설명한 TLS 표준을 준수하여 구현된 오픈소스 라이브러리로 mbedTLS, OpenSSL, wolfSSL 등이 있다. mbedTLS는 임베디드 아키텍처 회사 ARM이 주관하는 프로젝트로 경량 환경에서 효율적으로 동작할 수 있는 옵션을 제공하고 임베디드 환경에 맞게 다양한 운영체제와 호환된다.



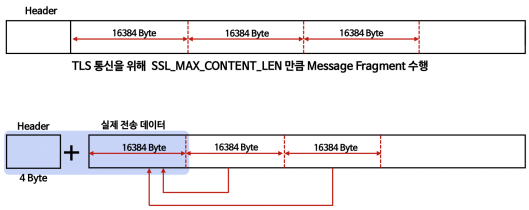
〔그림 2〕 TLS Cihpersuite 구조

Ⅲ. 구 현

이번 장에서는 전송할 데이터가 큰 격자기반 양자내성암호를 구현하기 위한 Handshake Message Fragmentation 기법과 새로운 Ciphersuite을 제안한다. TLS 표준에서는 프레임 버퍼의 크기를 16KB(16384 bytes)로 제한하고 있다. Message Fragmentation은 16KB 이상이 데이터 크기를 가지는 메시지를 나누어 보내는 기법이다. [그림 3]을 보면 TLS Handshake 과정을 마치고 암호화된 데이터를 송수신하는 Application Data 단계가 있다. mbedTLS 라이브러리는 이 단계에서 Message Fragmentation 기능은 있지만 Handshake에서는 지원하지 않는다. 따라서, 16KB 이상의 데이터 전송이 필요한 양자내성암호를 적용하기 위해서는 Handshake Message Fragmentation 기능을 구현할 필요가 있다. [표 1]을 보면 Lizard.KEM에서 16KB보다 큰 데이터 전송이 필요한 공개키와 암호문이 있다. 해당 파라미터는 TLS Handshake 과정을 나타내는 [그림 3]에서 Server/Client Key Exchange 과정 중 전송되어 데이터가 클 경우 Message Fragmentation 기능이 동작되어야 한다. [그림 4]와 같이 전체 데이터를 16KB만큼 나누고 데이터를 헤더와 함께 전송한다. 데이터를 전송하였으면 다음 16KB 크기의 데이터도 헤더와 함께 전송하는 방식으로 전체 데이터를 모두 전송한다. 송신측



〔그림 3〕 TLS Handshake 과정



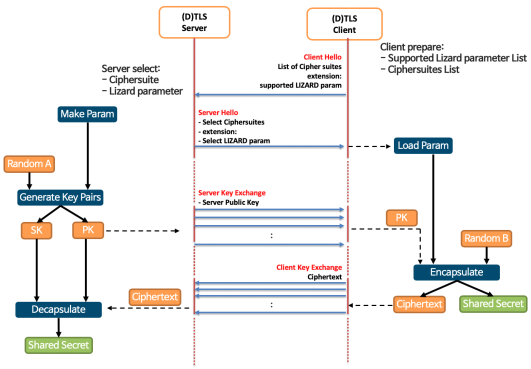
〔그림 4〕 Message Fragmentation 과정

에서 메시지를 분할하여 보내는 기능이 필요하듯이 수신측에서는 분할한 메시지를 모으는 Reassemble 기능이 있어야 한다. 수신측에서는 분할한 데이터에 헤더를 분리하여 실제 전송 데이터를 순차적으로 저장한다. 이와 같이 송수신측에 각각 Message Fragmentation과 Reassemble 기능을 구현하여 Handshake 과정에서 16KB 이상의 데이터 전송을 가능하게 구현하였다.

본 논문에서는 Lizard.KEM, RLizard.KEM을 적용하기 위해 새로운 Ciphersuite도 구현하였다. 서명 알고리즘으로 ECDSA, 블록암호는 ARIA128 CBC 운용모드, 해시함수는 SHA256을 사용하고 키 교환 알고리즘으로 Lizard.KEM, RLizard.KEM을 사용한다. 최종적으로 [그림 5]와 같이 Ciphersuite을 RLizard.KEM, Lizard.KEM 각각 0xCDFF, 0xCDFE로 구현하였다. [그림 6]은 TLS에서 Lizard 키 교환 알고리즘을 적용하여 동작하는 과정을 보여준다. 서버는 Server Key Exchange 단계에서 비밀키(SK), 공개키(PK)를 만들고 공개키를 클라이언트로 전송한다. 그러면 클라이언트는 Client Key Exchange에서 수신받은 공개키로 임의의

```
#define MBEDTLS_TLS_RLIZARD_ECDSA_WITH_ARIA_128_CBC_SHA256  0xCDFF
#define MBEDTLS_TLS_LIZARD_ECDSA_WITH_ARIA_128_CBC_SHA256   0xCDFE
```

〔그림 5〕 Lizard를 적용한 Ciphersuites



〔그림 6〕 TLS상에서 Lizard의 동작 과정

값을 캡슐레이트(Encapsulate)하여 공유키(Shared Secret)와 암호문(Ciphertext)을 만들고 암호문을 전송한다. 서버는 암호문을 수신받아 비밀키와 공개키로 디캡슐레이트(Decapsulate)하여 클라이언트와 동일한 공유키를 갖는다. Server/Client Key Exchange 과정 중에서 [표 1]에서 볼 수 있듯이 공개키와 암호문의 크기가 16KB 이상이면 Message Fragmentation 기능을 사용한다.

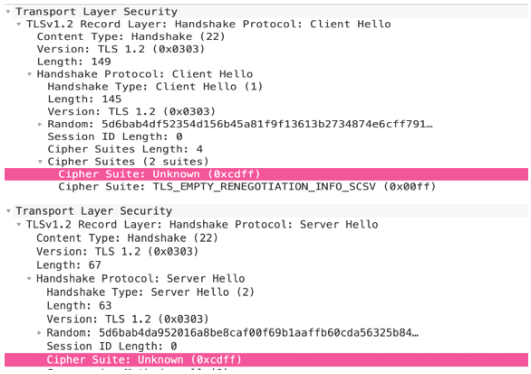
Ⅳ. 실 험

mbedTLS 상에 적용한 격자기반 양자내성암호 Lizard의 성능을 평가하기 위해 실험환경을 구축하였다. [표 2]는 성능평가 환경을 나타내고 있다. [그림 7]은 TLS 통신 과정 중에 Wireshark 프로그램을 통해 패킷을 캡처한 결과이다. TLS Handshake 과정 중 Client Hello 단계에서 RLizard.KEM을 적용한 ciphersuite(0xCDFF)을 선택하고 Server Hello 단계에서도 같은 ciphersuite으로 결정한 것을 볼 수 있다. [그림 8]은 Handshake Message Fragment 기능이 동작하는 패킷을 캡처한 내용을 보여준다. 메시지가 분할되어 TLS 프레임 버퍼의 최대 크기인 16384bytes와 헤더의 크기 4bytes를 합쳐 16388bytes 크기의 패킷이 여러 번 보내지는 결과를 확인할 수 있다.

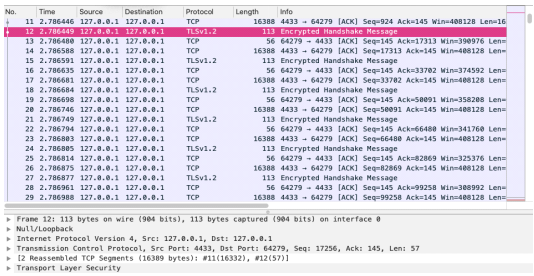
mbedTLS는 성능 평가하기 위한 벤치마크 도구를 제공한다. mbedTLS에서 제공하는 벤치마크는 네트워크 환경을 고려하지 않고 암호 알고리즘 자체의 성능을 평가한다. 벤치마크 결과의 단위는 handshake/s이고 선택한 암호 알고리즘으로 3초간 handshake를 수행하여 초로 나눈 결과이다. [표 3]은 기존 공개키 기반 알고리즘과 양자내성암호 Lizard를 적용한 키 교환 알고리즘의 벤치마크 결과이다. RLizard.KEM Category 1과 같이 128-bits의 안전도를 보장하는 ECDH Curve 25519와 비교하면 성능이 더 뛰어나지는 않지만 비슷한 성능을 보인다. [그림 9]의 결과는 Handshake의 Client

〔표 2〕 성능평가 환경

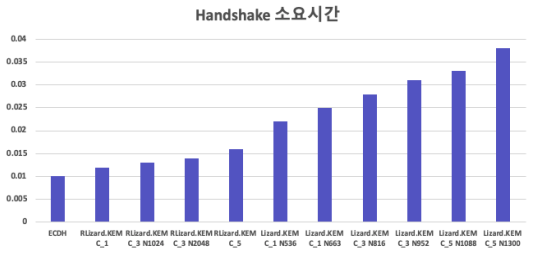
운영체제	macOS Catalina
CPU	Intel i7-6820HQ 2.70GHz
RAM	16GB
그래픽카드	AMD Radeon Pro 455
Wireshark	version 3.0.2



〔그림 7〕 Ciphersuite 선택 과정 캡처



〔그림 8〕 Message Fragment 패킷 분석



〔그림 9〕 Handshake 소요 시간

Hello부터 Application Data까지의 시간을 측정한 결과이다. 100번 수행하여 평균 값을 ms로 나타낸다. 같은 안전도를 보장하는 ECDH와 Lizard 알고리즘을 적용한 Category1을 비교하면 RLizard.KEM의 경우 유사한 성능을 보이지만 Lizard.KEM은 2배 이상의 시간이 소요된다.

양자내성암호를 적용한 RLizard.KEM의 경우 ECDH와 작은 성능 차이가 있었다. 이러한 실험 결과를 보았을 때 양자내성암호를 적용한 키 교환 알고리즘으로 기존의 알고리즘을 대체가 가능함을 볼 수 있다. 또한 전송 데이터가 큰 격자기반 알고리즘의 경우에도 Message Fragmentation 기능을 구현하여 실제 동작 가능함을 보였다.

〔표 3〕 키 교환 알고리즘의 성능평가 결과

(단위 : handshake/s)

알고리즘	성능
DH-2048	148
DHE-2048	69
ECDH Curve 25519	549
ECDHE Curve 25519	645
RLizard.KEM Category 1	471
RLizard.KEM Category 3 N1024	310
RLizard.KEM Category 3 N2048	261
RLizard.KEM Category 5	96
Lizard.KEM Category 1 N536	1
Lizard.KEM Category 1 N663	1
Lizard.KEM Category 3 N816	1
Lizard.KEM Category 3 N952	1
Lizard.KEM Category 5 N1088	1
Lizard.KEM Category 5 N1300	1

Ⅴ. 결 론

본 논문에서는 격자기반 양자내성암호를 mbedTLS 라이브러리에 적용한 결과를 제시하였다. mbedTLS의 경우 Handshake Message Fragmentation 기능이 구현되어 있지 않아 큰 데이터를 전송하는데 문제가 있었으나, 직접 송수신측에 각각 Fragmentation, Reassemble 기능을 구현하여 해결하였다. 이와 같은 결과를 제시하여 16KB 이상의 데이터 전송이 필요한 격자기반 양자내성암호가 적용 가능함을 보였다. 구체적인 예시로 Lizard.KEM, RLizard.KEM을 구현하였고 기존 키 교환 알고리즘과의 비교를 통해 성능 평가를 제시하였다. 이를 통해 mbedTLS 라이브러리에 격자기반 양자내성암호를 적용하여 양자컴퓨터에도 안전한 통신 환경을 구성하는 것이 가능하다는 것을 확인하였다. 향후 Lizard외의 다른 격자기반 양자내성암호 알고리즘을 적용함을 보이고 OpenSSL에서 구현한 결과들과 성능비교를 수행할 수 있다. 또한 Message Fragmentaion 기능이 필요한 Lizard.KEM의 경우 RLizard.KEM과 비교하여 느린 성능을 보인 것으로 보아 최적화 연구가 필요하다.

참 고 문 헌

[1] Arute, F., Arya, K., Babbush, R. et al, Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510, 2019.

[2] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput, 26(5), 1484-1509, 1997.

[3] D. Stebila and M. Mosca, Post-quantum key exchange for the internet and the open quantum safe project, in Conf. International Conference on Selected Areas in Cryptography, 14-37, 2016.

[4] M. Ajtai, Generating hard instances of lattice problem, in Conf. The 28th Annual ACM Symposium on Theory of Computing, 99-108, 1996.

[5] J. H. Cheon et al. Lizard Tech report[Online]. Available : https://csrc.nist.gov/Projects/post-quantum-cryptography/Round-1-Submissions

[6] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, Journal of the ACM(JACM), 56(6), 1-40, 2009

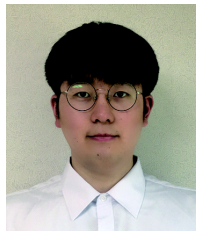
[7] J. Alwen et al. Learning with Rounding," CRYPTO 2013, 57-74, 2013

[8] M. Abe et al. Tag-KEM/DEM: A new framework for hybrid encryption, Journal of Cryptography, 21(1), 97-130, 2008

〈저 자 소 개〉

박 찬 희 (Chanhui Park)

2018년 2월 : 대구대학교 컴퓨터정보공학부 졸업
2020년 2월 : 부산대학교 전기전자컴퓨터공학과 석사
2019년 11월~현재 : 스마트엠투엠 연구원
<관심분야> 암호학, 정보보호, IoT 등



윤 영 여 (Youngyeo Yun)

2020년 2월 : 부산대학교 전기컴퓨터공학부 졸업
2020년 3월~현재 : 부산대학교 전기전자컴퓨터공학과 석사과정
<관심분야> 정보보호, 암호학 등

박 해 룡 (Haeryong Park)

1999년 2월 : 전남대학교 수학과 졸업
2001년 2월 : 서울대학교 수리과학부 석사
2006년 8월 : 전남대학교 정보보안협동과정 박사
2000년 12월~현재 : 한국인터넷진흥원 수석연구원
<관심분야> 암호기술 설계 및 분석, 양자내성암호, 동형암호, 보안프로토콜 등

최 은 영 (Eunyoung Choi)

2001년 8월 : 고려대학교 수학과 졸업
2003년 8월 : 고려대학교 정보보호학과 석사
2009년 8월 : 고려대학교 정보보호학과 박사
2007년 10월~현재 : 한국인터넷진흥원 책임연구원
<관심분야> 암호프로토콜, IoT, 양자내성암호 등

김 호 원 (Howon Kim)

1993년 2월 : 경북대학교 전자공학과 졸업
1995년 2월 : 포항공과대학교 전자전기공학과 석사
1999년 2월 : 포항공과대학교 전자전기공학과 박사
2008년 3월~현재 : 부산대학교 전기컴퓨터공학부 교수
<관심분야> 암호학, 정보보호, 블록체인, IoT 등