공학전문석사 학위 연구보고서

국방 환경에 적합한 양자내성 암호 설계 및 구현

2020년 2월

서울대학교 공학전문대학원

응용공학과 응용공학전공

정 치 곤

공학전문석사 학위 연구보고서

국방 환경에 적합한 양자내성 암호 설계 및 구현

2020년 2월

서울대학교 공학전문대학원

응용공학과 응용공학전공

정 치 곤

국방 환경에 적합한 양자내성 암호 설계 및 구현

지도교수 백 윤 흥

이 프로젝트 리포트를 공학전문석사 학위
연구보고서로 제출함

2020년 2월

서울대학교 공학전문대학원

응용공학과 응용공학전공

정 치 곤

정치곤의 공학전문석사 학위 연구보고서를 인준함

2020년 2월

위 원 장: _____

위 원: _____

위 원: _____

국문초록

양자컴퓨터로 인한 공개키 암호 해독 위협이 가시화 되면서, 양자내성 암호 표준화 공모가 미국 표준기술연구원(National Institute of Standards and Technology, NIST)를 중심으로 이뤄지고 있다. 반면, 군에서는 양자내성 암호 도입이 더욱 시급함에도 불구하고 관련 연구가 미진한 실정이다.

민간에서는 NIST 주관 공모전을 통해 선택된 국제표준 알고리즘을 사용하면 소프트웨어 업데이트 또는 장비의 교체 과정만 감내하면 양자내성 암호로 전환할수 있지만, 군은 그렇지 못하다. 왜냐하면, 국방 환경에서는 다양한 무기 및 통신체계에 적용될 알고리즘이 필요하기 때문에 성능, 대역폭, 정확성 및 안전성 측면에서 보다 엄격한 기준이 요구된다. 하지만, 현재까지 NIST 주관 표준화 공모에 제출된 후보 알고리즘들은 성능, 대역폭, 정확성 및 안전성 측면에서 저마다의 장단점을 갖고 있어 모든 측면에서 탁월한 알고리즘 하나를 선택하기 어렵다. 또한, 군사작전간 분실 및 피탈 상황에서도 암호장비(또는 소프트웨어) 내부의 비밀 키 등이 탈취되지 않도록 부채널 공격에도 대응해야 하지만 방어대책이 부족하다. 게다가, 국산 암호만을 군사 비밀 보호에 사용할 수 있는 국내 법규도 향후 NIST에서 선정한 국제표준 알고리즘을 사용할수 없는 제한사항이 된다.

따라서, 이 연구에서는 국방 환경에 적합한 양자내성 암호 알고리즘을 설계하고 성능을 평가한다. 본 연구에서 설계한 알고리즘은 LizarMong이라고 명명했으며, 국산 양자내성 암호 알고리즘인 RLizard [1]를 기반으로 개선한 체계이다. LizarMong은 128bit 보안강도인 `Comfort`와 256bit 보안강도인 `Strong` 버전으로 사용할수 있으며, 공격자가 평문을 선택할수 있는 상황 하에서의 구별 공격(Indistinguishability under chosen-plaintext attack, IND-CPA)에 안전한 공개 키 암호 알고리즘(Public-key Encryption, PKE)와 가장 강력한 공격자 가정인 적응형 선택 암호문 상황하 구별 공격(Indistinguishability under adaptive chosen ciphertext attack, IND-CCA2)에도 안전한 키 교환 매커니즘(Key Encapsulation Mechanism, KEM)을 지원한다.

LizarMong의 기반이 된 RLizard는 우수한 성능과 안정적인 보안성과 정확성을 갖고 있지만, 국방 환경에 사용하기에는 대역폭이 크고 부채널 공격에 비교적 많이 노출된 단점이 있다. LizarMong은 RLizard의 문제점을 해소하기 위해 NIST 주관 표준화 공모 후보 알고리즘들의 장점들을 조합하고, 최신 부채널 공격 대응기법 등을 종합하여 설계하였다. 구체적으로, 대역폭 감소를 위해 Ring-Learning With Error(RLWE)에 사용되는 법(modulus)인 q를 2^8으로 대폭 감소시키고 암호문 및 공개키 압축 기법을 적용하였다. 이로 인해 감소하는 알고리즘의 정확성을 보상하기 위해 오류 정정 부호화 기법인 XE5 [2]를 도입했다. 또한, 부채널 공격 저항성을 갖기 위해 공격 표면(attack surface)을 줄이고 대응 기법을 삽입하였다. 공격 표면을 줄이기 위해 알려진 캐시 공격 및 타이밍 공격들이 취약점으로 악용한 법 연산(modulus operation)과 오류 샘플링에서의 표 탐색 기법을 제거하였다. 즉, 법(modulus)을 2의 지수 승으로 선택하고 에러 샘플링 방법을 기존의 누적확률분포표(Cumulative Distribution Table, CDT)를 탐색하는 방법에서 중심이항분포(centered binomial distribution) 계산 방법으로 변경하였다. 또한, 다항식 곱셈 연산과 에러 및 비밀값 생성 과정을 악용하는 오류주입 및 전력분석 공격 대응을 위한 방어기법을 내재하였다.

결과적으로 LizarMong은 알려진 여러 부채널 공격으로부터 저항성을 갖을 뿐만 아니라, RLizard와 유사한 정확성과 안전성 수준에서 최대 85% 작은 대역폭과 3.3배 빠른 성능을 보인다. NIST 주관 표준화 공모 후보 알고리즘들과 비교해도 유사한 정확성과 안전성을 갖고도 대역폭은 약 5-42% 작으며, 성능은 약 1.2-4.1배 빠르다.

따라서, LizarMong은 성능, 대역폭, 정확성과 안전성 모든 측면에서 우수할 뿐만 아니라 부채널 공격에 저항하며, 국내에서 설계된 알고리즘이므로, 국방 환경에 적합하게 사용할 수 있다.

주요어: Ring learning with error, post-quantum cryptography, 양자내성 암호

학 번: 2018-26518

목 차

국문초록 **i**

제 1 장 서 론 **5**

제 1 절 연구의 중요성 5

제 2 절 연구 개요 7

제 2 장 배경 지식 **10**

제 1 절 기호 10

제 2 절 격자 기반 문제 10

1. Ring Learning With Error 10

2. Ring Learning With Rounding 11

제 3 절 RLWE 기반 알고리즘 11

제 4 절 NIST 주관 표준화 공모 후보알고리즘 14

제 5 절 RLizard 16

제 3 장 국방 환경에 적합한 양자내성 암호 **18**

제 1 절 설계 요소 선택 19

제 2 절 상세 알고리즘 23

1. IND-CPA PKE 23

2. IND-CCA2 KEM 24

제 3 절 파라미터 25

제 4 절 안전성 분석 26

1. IND-CPA 및 IND-CCA2 증명 26

2. 알려진 공격에 대한 안전성 분석 28
제 5 절 정확성(복호화 실패 확률) 29
제 6 절 부채널 공격 30

제 4 장 평가 **34**

제 5 장 결 론 **38**

표 목 차

표 3.1 보안 강도에 따른 세부 파라미터 26
표 3.2 공개키, 비밀키 및 암호문의 크기(단위: bytes) 26
표 3.3 파라미타 별 복잡도 평가 결과 29
표 3.4 복호화 실패 확률 30
표 3.5 알려진 부채널 공격과 LizarMong의 대응기법 32

표 4.1 NIST 공모 2라운드 후보 알고리즘 및 RLizard의 KEM과의 평가 비교표 35

그 림 목 차

그림 2.1 RLizard IND-CCA2 KEM Algorithm 16

그림 4.1 KEM 기준 대역폭 및 성능 비교 (좌) 128bit 보안 강도 (우) 256bit 보안 강도 (참고: ● 는 IND-CCA2에서 안전성과 각 보안 수준과 유사한 정확성을 갖는 알고리즘이며, ×는 그렇지 못한 경우를 표시하였다.) 34

제 1 장 서 론

제 1 절 연구의 중요성

충분히 큰 규모의 양자컴퓨터가 등장하면 Shor's 알고리즘을 활용하여, 현재의 컴퓨터 환경에서 어려운 문제로 잘 알려진 인수분해와 이산대수 문제를 다항식 시간안에 해결할수 있다고 잘 알려져 있다. 인수분해와 이산대수 문제는 RSA, ECC와 같은 현대 공개키 암호의 안전성을 보장해주는 기반으로 두 문제를 쉽게 해결할수 있으면 RSA와 ECC도 해독된다. 공개키 암호는 우리 삶 속에 녹아있는 인터넷 보안, 인증 기술, 블록체인 등의 기술에 기밀성, 무결성, 인증, 부인방지의 보안 기능을 제공하는 기반이기 때문에 해독 가능성이 제기되는 것 만으로도 심각한 보안 위협이다.

양자내성 암호(post-quantum cryptography)는 양자컴퓨터에 의한 보안 위협에 대응하는 새로운 암호체계로서 현재의 컴퓨터 환경은 물론이고, 양자컴퓨터 환경에서도 안전한 새로운 공개키 암호체계이다. 즉, 이산대수와 인수분해처럼 양자컴퓨터에서 쉽게 해결되는 문제 대신에 격자(Lattice), 부호(Code) 이론, 다변수 다항식(Multivariate) 문제와 같이 양자컴퓨터와 현재의 컴퓨터로도 효율적인 해결 방법이 없는 것으로 알려진 어려운 문제를 기반으로 암호를 설계한다.

비록 현대 공개키 암호에서 사용하는 큰 크기의 인수분해 및 이산대수 문제를 해결할 만큼 충분히 큰 규모의 양자컴퓨터에 대한 정의와 등장 시기에 대한 이견이 분분하지만, Mosca's inequation [3]에 따르면 양자내성 암호의 연구와 상용화는 매우 시급함을 인정할 수 있다. Mosca's inequation은 비밀의 보호기간(x)과 새로운 암호체계가 상용화 되는데 필요한 기간(y)의 합이 양자컴퓨터 등장 시점(z)보다 작거나 같아야 한다는 의미이고, y는 15년, z는 15 20년으로 가정하였다. 이 등식을 기초로 현재 공인인증서 및 인터넷 트래픽 보안에 사용되는 RSA-2048bit 보안 수준의 공개 키 암호는 2027년까지 1/6의 확률로, 2032년까지는 1/2의 확률로 해독될

것으로 예측하고 있다. 이처럼 양자내성 암호 연구 및 상용화의 중요성이 커지자 미국 국립표준기술연구소 (National Institute of Standards and Technology, NIST)에서는 늦어도 2024년까지 표준 제정을 목표로 2016년부터 양자내성 암호 표준화 공모를 진행하고 있으며 2019년 1월 31일 1라운드 경쟁을 마치고 현재는 24개의 알고리즘이 2라운드 경쟁을 치르고 있다.

국방 환경에서 Mosca's inequation을 적용하면 양자내성 암호의 도입이 더욱 시급함을 알수 있다. 첫째, 군사비밀은 민간에서의 중요정보보다 비밀의 보호기간(x)이 길다. 군사비밀은 필요성에 따라 짧게는 1년, 길게는 30년 또는 영구적으로 보존하고 있다. 만약 암호화되어 있는 비밀이 유출된다면 당장은 해독할 수 없겠지만, 그 비밀이 유효한 기간 내에 양자컴퓨터가 개발되어 해독된다면 적(adversary)에게 가치 있는 정보를 내어주는 꼴이 된다. 둘째, 국방예산과 지금의 암호체계 운영 방식을 고려하면 민간보다 암호체계 배포 소요시간(y)이 짧다고 단정짓기 어렵다. 국방 환경은 제한된 사용자가 암호체계를 운용하기 때문에 민간에서의 배포 작업보다 빠르게 조치 될 것으로 생각할수 있지만, 90년대 초반에 개발된 암호장비가 30년 가까이 사용되고 있는 현실과 다양한 통신 및 무기체계와의 호환성 검토, 정책 결정 및 예산 획득 절차 등을 고려시 민간보다 확연히 짧다고 속단할 수 없다. 결론적으로, 양자컴퓨터의 등장시점(z)이 같더라도, 비밀의 보호기간(x)이 길고, 암호 배포에 필요한 기간(y)이 비슷하기 때문에, 국방 환경에서 양자내성 암호 도입은 더욱 시급하다. 안타깝게도, 현재까지의 군 암호체계는 대칭키 위주로 운영되었기 때문에 국방 환경에 적합한 양자내성 암호에 대한 연구는 미진한 실정이다. 하지만 첨단과학기술군을 목표로 드론, 로봇과 같은 첨단 장비들과 IoT를 주축으로하는 초연결네트워크를 구성하기 위해서는 많은 수의 소형장비들이 도입되고 이런 장비들에 대한 암호키 관리 및 인증, 부인방지 등 다양한 보안서비스를 제공하려면 군도 공개 키 암호를 사용할 여건이 조성될 것이다. 또한, 암호 알고리즘은 충분한 검증 시간이 필요한 특성이 있기 때문에 지금 시점에서의 양자내성 암호 알고리즘 설계 연구는 매우 적절하다고 볼수 있다.

제 2 절 연구 개요

본 연구보고서에서는 국방 환경에 적합한 양자내성 암호 알고리즘을 설계하고 구현한다. 국방 환경은 민간에서보다 더 높은 보안성을 요구하며, 최근 화두가 되는 드론 및 IoT에도 적용될수 있도록 경량화 되어야 한다. 뿐만 아니라, 적에게 탈취되거나 분실하는 상황에서도 암호장비에 포함된 비밀 정보를 획득할 수 없도록 각종 부채널 공격(side-channel attack)에도 안전해야 한다.

NIST에서 추진 중인 양자내성 암호 알고리즘의 여러 표준화 후보들 중, Ring Learning With Error (RLWE) 계열* 알고리즘은 안전성이 잘 증명되어 있고, 다른 계열(부호 기반, 다변수 다항식 기반)에 비해 대역폭과 성능이 우수하여 키 설정 매커니즘(Key Encapsulation Mechanism, KEM) 및 공개 키 암호(Public-key Encryption, PKE) 분야에서 각광 받고 있다.

RLWE 계열 알고리즘의 설계 요소는 크게 격자의 차원(dimension)과 모듈러스(modulus), 오류의 크기와 모듈러스의 비율인 오류비율(error rate) 그리고 Ring의 구조 및 기반 문제로 볼 수 있다. 이 요소들은 상충관계를 가지며, 알고리즘의 안전성과 정확성(복호화 실패 확률), 성능(연산 속도), 대역폭(공개 키+암호문 크기)을 결정한다. 설계 요소 간의 상충관계를 관점으로 NIST 표준화 공모 2라운드에 진출한 RLWE 계열 알고리즘을 살펴보면 몇 가지로 크게 구분 지을수 있다. 우선 기반 문제에 따라 알고리즘이 갖는 태생적 기질이 결정된다. RLWE를 기준으로 본다면 RLWR은 에러샘플링 과정이 없고 하위 bit를 제거하기 때문에 성능과 대역폭이 우수하다. MLWE 및 MLWR은 상대적으로 작은 차원을 사용할 수 있기 때문에 RLWE에 비해 대역폭을 줄일 수 있다. 상대적으로 오랜 시간 연구가 진행되었고 ideal lattice 기반 난제로부터 어려움이 증명된 RLWE가 다른 두 기반 문제보다 보수적인 안전성을 갖고 있다고 주장할수 있지만, RLWR과 MLWE(R)도 각각 증명된 안전성을 갖고 있기 때문에 그 차이를 문제삼기 힘들다. 태생적 기질은 다른 설계 요소를 통해

*Ring-LWE (RLWE), Ring learing with rounding (RLWR), Module-LWE (MLWE), Module-LWR (MLWR), Integer-MLWE (IMLWE)

후천적으로 보완되는 경향을 보이나, NIST 2라운드 알고리즘 중 가장 작은 대역폭을 보이는 것이 RLWR 기반의 Round5[2]란 점에서 그 특징을 무시할 수 없음을 알 수있다. 또 하나의 두드러진 구분은 모듈러스의 크기이다. 큰 모듈러스를 갖는 NewHope[4], KYBER[5], SABER[6]와 작은 모듈러스를 갖는 LAC[7], Round5로 구분할 수 있다. $2^{12\sim14}$ 수준의 큰 모듈러스를 갖는 NewHope 등은 비교적 큰 에러를 섞어서 안전성을 확보해도 큰 모듈러스 덕분에 에러비율이 작게 유지되어 정확성을 만족할 수 있다. 큰 모듈러스에 따른 연산량 및 대역폭 증가 문제는 고속 곱셈 알고리즘(예: NTT, Toom-cook)을 사용해 계산복잡도를 줄이고 암호문 및 공개 키 압축 기법을 사용하여 보완했다. 반대로 $2^{8\sim12}$ 수준의 작은 모듈러스를 갖는 LAC 등은 상대적으로 적은 대역폭과 우수한 연산속도를 갖을수 있다. 하지만, 큰 모듈러스에서 사용하는 크기의 에러를 그대로 사용하면 에러비율이 커져서 정확성이 낮아지는 문제가 발생한다. 따라서, 이진 또는 삼진 크기의 매우 작은 에러를 섞고 오류 정정 부호화 기법을 사용해 보정함으로써 안전성과 정확성을 유지한다. 이처럼 2라운드 후보 알고리즘들은 각 기반 문제 또는 모듈러스 등의 선택에 따라 설계 요소 간의 상충관계를 보완하기 위해 저마다의 기법을 사용했지만 안전성, 정확성, 성능, 대역폭이 모두 뛰어난 알고리즘 하나를 꼽지는 못하는 실정이다.

한편, NIST 표준화 공모는 RLWE 계열 암호에 대한 연구를 촉진시켜 최근 다양한 결과가 공개되고 있다. 특히, 공격자 관점에서 부채널 공격 및 대응기법이 다양하게 등장하고 있으며, 그 동안 설계한 RLWE 계열 알고리즘이 정확성 분석에 공통적으로 사용한 오류 발생의 bit 간 독립성 가정이 반증되는 연구[8]도 발표되어 고려가 필요한 시점이다. 하지만, 현재까지 알고리즘들은 이러한 최신 연구를 반영한 알고리즘 설계는 미흡한 상황이다. 특히, 부채널 공격의 경우 고속 곱셈 알고리즘이나 효율적인 에러샘플러 구현 등 설계 요소 간의 상충관계를 보완하기 위한 최적화 기법을 악용한 경우들이 많다. 이는 RLWE 계열 성능평가에 중요한 영향을 줄 수 있다. 예를 들면, NTT 공격 대응을 위해 이를 제거하면 계산복잡도가 크게 증가하고, 대응기법을 넣게 되면 NTT로 얻는 성능 이득이 줄어들기 때문이다. 따라서,

부채널 공격 대응기법을 알고리즘에 내재하여 새롭게 성능을 평가할 필요가 있다.

이 연구에서의 알고리즘은 LizarMong으로 명명하였다. LizarMong은 국내에서 개발하여 NIST의 표준화 공모 1라운드까지 진출했던 RLizard[1]에 기반을 둔다. 일반적으로 암호 알고리즘의 안전성은 공개 검증을 거칠 때 보다 신뢰할수 있다는 점에서 국방 환경에서의 높은 보안성을 만족하기 위한 선택이다. RLizard를 국방 환경을 고려하여 개선하기 위해 RLizard 단점을 분석하고 NIST 표준화 공모 2라운드 후보들을 면밀히 분석하여 단점을 보완할 수 있는 기법들을 적용하여 보완한다. 또한 최근 연구들을 조사하여 안전성과 효율성을 더욱 배가 시킨다. 알고리즘의 구현은 소프트웨어(C 언어) 구현을 통해 NIST 표준화 공모 2라운드 후보알고리즘 및 RLizard와 성능을 비교한다. 이 구현은 하드웨어 형태의 암호장비를 사용할 수 없는 환경에서 사용될 수 있는 좋은 참조가 된다. 또한 FPGA 구현을 통해 기존의 암호장비를 대체할 수 있는 가능성을 제시하고 보다 나은 성능을 제공한다.

주요 기여사항 국방 환경에 적합한 양자내성 공개 키 암호 및 키 교환 매커니즘인 LizarMong을 설계 및 구현하였다.

- 대역폭을 줄이기 위해, 모듈러스를 작게(2^8) 설정하고 NIST 2라운드 후보들이 공통적으로 사용한 암호문 압축 및 공개키 압축 기법을 사용하였다.
- 작은 모듈러스 설정으로 인해 감소한 알고리즘 정확성 보완을 위해 오류 정정 부호화 기법(error correcting code)인 XE5 [2]를 적용하였다.
- 부채널 공격 저항성을 갖기 위해 고속 희소 다항식 곱셈에 하이딩(hiding) 기법을 적용하도록 고안하였다. 또한 누적분포표(Cumulative Distribution Table, CDT) 기법을 사용한 에러 샘플러 대신 중심이항분포 샘플러를 사용한다.
- 알고리즘의 정확성을 더욱 정확하게 평가하기 위해, 각 bit별 오류 발생의 비독립성을 가정하여 복호화 실패확률을 계산하였다. 따라서, 복호화 실패를 악용한 공격에 더 강인하다.

제 2 장 배경 지식

제 1 절 기호

이 연구보고서에서 사용하는 log는 별도로 표시하는 경우를 제외하고 밑이 2인 logarithm이다. $\mathbb{Z}_q$는 $\mathbb{Z} \bigcap (-q/2, q/2]$를 의미하며 q는 양의 정수이다. R_q는 $X^n + 1$을 모듈러스로 갖고 계수를 $\mathbb{Z}_q$로 하는 다항식의 환(Ring)으로 $\mathbb{Z}_q[X]/(X^n + 1)$이다. 다항식은 굵은 글씨체(예: $\mathbf{a}$)로 표기하며, R_q에서의 곱셈 연산은 $*$로 표기한다. $\lfloor r \rceil$은 실수 r에 가장 가까운 정수로의 반올림이며 $\lfloor \mathbf{a} \rceil$는 다항식 $\mathbf{a}$의 각 계수에 대해 가장 가까운 정수로 반올림하는 연산이다. norm은 모두 2-norm이고 $\|x\|$로 표기한다. $x \parallel y$는 x와 y를 연접(concatenate)하는 것이다. 알고리즘 설계에 사용되는 분포(distribution)는 두 가지로, $HWT_n(h)$와 ψ_{cb}이다. $HWT_n(h)$는 $\{-1, 0, 1\}^n$의 부분집합 중 0이 아닌 값이 h개인 원소들의 집합에서 유니폼하게 샘플링한 분포이다. ψ_{cb}는 중심은 0이고, 표준편차는 $\sqrt{cb/2}$인 중심이항분포이다. $\ll$와 $\gg$는 각각 왼쪽으로의 비트 이동과 오른쪽으로 비트 이동이다. $\texttt{SHAKE256}(m, len)$는 스폰지 구조 해시 함수로 m을 입력으로 받아 len 길이의 해시 값을 출력하는 함수이다. eccENC와 eccDEC는 각각 오류 정정 부호화 기법을 사용하여 인코딩 및 디코딩을 수행하는 함수이다.

제 2 절 격자 기반 문제

1. Ring Learning With Error

양의 정수 n과 q, 그리고 n차 기약 원분다항식(irreducible cyclotomic polynomial) $f(X) \in \mathbb{Z}$이 주어졌을 때, 환 $R_q := \mathbb{Z}_q[X]/(f(X))$을 정의한다. χ를 R_q에서 e가 작은 계수를 갖도록 샘플링하는 분포라 하고, D는 R_q에서의 s의 분포라 하자. $\mathbf{a}$

는 R_q에서 유니폼 랜덤하게 선택한다고 하자. Decision Ring-LWE문제는 R_q^2에서 유니폼 분포와 $(\mathbf{a}, \mathbf{a} * \mathbf{s} + \mathbf{e}) \in R_q^2$를 구별하는 문제이고, Search Ring-LWE문제는 $(\mathbf{a}, \mathbf{a} * \mathbf{s} + \mathbf{e}) \in R_q^2$가 주어졌을 때 s를 찾는 문제이다. Decision Ring-LWE문제와 Search Ring-LWE문제는 ideal lattice 위에서 어려운 문제인 approximate shortest vector problem (approximate SVP)과 shortest independent vectors problem (SIVP)보다 어렵다 [9].

2. Ring Learning With Rounding

Ring-LWE 문제에서 에러를 결정적으로 생성하기 때문에 Ring-LWE문제의 derandomize 버전이라고도 볼수 있다. 다시 말해, Ring-LWE에서 $\mathbf{e}$를 더하는 대신 $p(< q)$로 모듈로 연산을 함으로써 하위 bit를 0으로 바꾸어 에러 $\mathbf{e}$를 더하는 효과를 내는 방식이다. Decision Ring-LWR문제는 $R_q \times R_p$에서 유니폼 분포와 $(\mathbf{a}, \lfloor (p/q) \cdot \mathbf{a} * \mathbf{s} \rceil) \in R_q \times R_p$를 구별하는 문제이다. Ring-LWR 문제는 Ring-LWE 문제만큼 어렵다 [10, 11].

제 3 절 RLWE 기반 알고리즘

최초의 LWE 기반 공개키 암호 알고리즘 [12]은 행렬 및 벡터의 내적 연산, 가우시안 분포에서의 에러 샘플링 등으로 인해 성능이 느리고 대역폭도 큰 문제를 갖고 있어서 후속 연구들은 이 문제를 해소하는 방향으로 진행되었다. 첫 번째 방법은 격자에 Ring 구조를 가미한 Ring-LWE [9] 이다. Ring-LWE는 안전성이 증명되었고 LWE에 비해 성능과 대역폭에서 큰 이득을 볼 수 있어서 폭넓게 사용되고 있다.

Ring-LWE의 발전과정을 성능 및 대역폭 향상을 위한 노력 측면에서 살펴보면 크게 다항식 곱셈 연산의 고속화와 에러 및 비밀 샘플링의 단순화로 볼수 있다.

먼저, 다항식 곱셈은 시간복잡도가 $O(n^2)$이기 때문에 행렬 및 벡터간의 연산과 복잡도가 같아 차이가 없다고 볼수 있다. 하지만, 다항식 곱셈은 Number-theoretic

transform (NTT), Toom-cook, Karatsuba 또는 희소(sparse) 다항식 곱셈과 같이 대수학에서 잘 연구된 고속 알고리즘을 사용할 수 있어서 시간복잡도를 최대 $O(n \log n)$까지 줄일 수 있다. NewHope [4], KYBER [5]는 NTT를 사용하는 대표적인 예이다. 또한, Ring을 $\mathbb{Z}_q[X]/(X^n+1)$로 하면서 q를 2의 지수 승으로 한정하면 다항식 모듈러 연산에서 성능 이득을 볼 수 있고, 안전성에 대한 연구가 잘 되어 있어 보안측면에서도 보수적 접근이 가능하다. 하지만, 이 방법은 n을 선택하는데 제약을 두기 때문에 달성하고자 하는 보안 수준(128bit, 192bit, 256bit)을 정확하게 맞출 수 없어서 불필요한 대역폭 낭비를 초래한다. 즉, n이 500-900 정도면 128-256bit 보안강도를 만족시킬 수 있는데, 2의 지수 승으로 사용하게 되면 512 다음이 1024이므로 적절한 n을 선택할 수 없다. Round5 [2]는 이 문제를 해결하기 위해 Ring을 $\mathbb{Z}_q[X]/(X^{n+1}-1)$을 사용한다. 이러한 Ring은 $n+1$이 소수면 되기 때문에, n 선택에 있어서 2의 지수 승보다 훨씬 자유로울수 있다. 따라서 대역폭을 절감할수 있다. 성능의 단점을 해소하기 위해 Round5는 lift/unlift 기법을 사용한다. 즉 $\mathbb{Z}_q[X]/(X^{n+1}-1)$ 연산을 하기 전 다항식에 $X-1$을 곱하여 lift하면, $\mathbb{Z}_q[X]/(X^n-1)$로 변환되기 때문에 간단히 모듈러스 연산을 할수 있다. 이후 다시 $X-1$을 나누어 unlift하면 원래의 Ring이 된다. 물론, 여기에도 성능 오버헤드가 있지만 복잡한 다항식 모듈러스 연산보다는 성능 하락을 최소화하고 대역폭을 줄인 좋은 예시이다.

두 번째로는 에러 e와 비밀 s를 가우시안 및 유니폼 분포 대신 보다 효율적인 하나의 분포에서 샘플링하는 것이다. 이 방법은 본래 정의된 RLWE 문제를 변형하는 것이므로 보안성에 영향을 주기 때문에 신중한 접근과 안전성 증명이 필요하다[13]. NewHope [4]는 중심이항분포(Centered Binomial Distribution)를 사용하면 RLWE 문제의 안전성에 영향을 주지 않는다는 것을 증명하였고, 이것을 인용하여 NIST 2 라운드 후보 알고리즘은 대부분 중심이항분포로 샘플링을 수행한다. 특히, 이후의 후속 연구들은 [5, 6] RLWE에 대한 알려진 공격들이 에러 및 비밀 분포의 종류가 아니라, 표준 편차에 의해 결정된다는 것을 주장하고 있어 중심이항분포는 일반적으로 잘 받아들여지고 있다. 한편, 에러를 생성하여 더하는 대신 하위 2-3bit를 버려서

에러를 만드는 방법인 Learning With Rounding (LWR) [10] 으로 변형하는 방법도 에러를 조정하는 맥락에서 발전하였다. LWR은 에러샘플링 과정이 없어 LWE에 비해 연산속도가 빠르고, 하위 2-3bit를 버리기 때문에 대역폭을 줄일 수 있다.

RLWE 기반 알고리즘의 또다른 발전 방향으로 알고리즘의 정확성을 향상시키기 위한 노력이 있다. 제 2장 제 2절에서 언급한 바와 같이 RLWE는 에러를 더하여 안전성을 단단히 하는 문제이다. 여기서 에러가 커지면, 좀 더 상세히 말하여 Ring의 계수 모듈러스인 q와 에러 분포의 표준편차로 구성되는 에러 비율이 커지면, 안전성은 올라가지만 복호화 실패확률도 늘어나 정확성이 떨어진다. 그래서 대부분의 연구는 q를 크게하고 에러도 적정한 크기를 유지하여 안전성과 정확성을 조율해 왔다. 하지만, 통신 선로상에서 발생하는 오류를 정정하기 위한 기법으로 잘 알려진 오류 정정 부호화 기법(Error-correction code)이 RLWE 계열 알고리즘의 대역폭 및 정확성, 안전성 향상에 도움이 된다는 연구 결과 [14]가 알려지면서 많은 알고리즘들이 채택하기 시작하였다. 특히, LAC [7]은 오류 정정 부호화 기법 (Error-correction code)을 사용하여 q를 251까지 줄였음에도 감내할 수준의 정확성을 달성했다. 이로 인해 모듈러스를 대폭 줄이는데 성공하였다.

이처럼, RLWE 기반 알고리즘은 성능, 대역폭, 정확성 그리고 안전성을 향상시키기 위해 많은 발전을 해왔고 지금도 관련된 연구가 꾸준히 발표되고 있다. 하지만, 각 기법들은 저마다의 장단점이 있고 각 요소들은 Trade-off 관계를 갖어 모든 면에서 훌륭한 알고리즘을 설계하기는 어려운 실정이다. 또한 각 기법들을 악용한 공격들도 많이 소개 되고 있다. 예를들면, NTT를 악용하는 부채널공격 [15]과 에러 정정 부호화 기법을 노린 오류 발생 종속성 [8] 및 복호화 실패확률 악용 공격 [16]이 대표적이다. 따라서 각각의 세부기술을 적절히 조합하고 이것들을 악용한 공격들에 대비하는 종합적인 설계방법이 요구된다.

제 4 절 NIST 주관 표준화 공모 후보알고리즘

FrodoKEM FrodoKEM [17]은 표준화 후보 중 유일하게 Ring구조를 사용하지 않는 LWE를 기반으로 설계되었다. 선택한 기반 문제에서 알 수 있듯이 가장 보수적인 안전성을 추구한다. 따라서, 에러 샘플링 방식도 가우시안 분포를 따른다. 이로 인해 다른 후보 알고리즘들보다 대역폭(암호문과 공개키 크기)이 크고 연산속도(암호화 및 복호화 시간)도 느리다. 하지만, 보수적인 안전성 덕분에 장기간 보안성능이 요구되는 국가기관이나 군에서 활용 가치가 있고, 저자들은 위성통신과 같이 한번 암호체계를 설치하면 변경이 어려운 응용환경에 적합하다고 주장하고 있다.

CRYSTALS-KYBER KYBER [5]는 Module-LWE 문제를 기반으로 만들어졌다. Module-LWE는 LWE와 RLWE의 사이에서 보안성과 효율성을 적절히 조합한 기반 문제로 평가된다. 따라서, RLWE에 비해 부족한 효율성을 보완하기 위해 다항식 곱셈에 NTT를 사용한다. 에러샘플링도 가우시안 대신 중심이항분포(Centered Binomial Distribution)를 선택하였다. 그 결과 후보 알고리즘 중 빠른 속도를 보이는 것으로 평가된다.

SABER SABER [6]는 Module-LWR 문제를 기반으로 만들어졌다. 파라미터 선정 등 기본적인 설계 개념은 KYBER와 비슷한 구조를 갖는다. LWR 계열이 갖는 장점 덕분에 같은 Module 구조를 갖는 KYBER보다 성능과 대역폭에서 효율적이다. Rounding을 사용해야 함에 따라 효율적인 Modular 연산이 필요하다. 따라서 KYBER와 달리 다항식 계수의 모듈러 값을 2의 지수 승으로 선택하였다. 이것은 NTT 고속 곱셈 기법을 사용할 수 없게 만든다. 하지만 RLWE에 비해 부족한 성능을 보완할 필요가 있기 때문에 차선책으로 Toom-Cook 고속 곱셈 기법을 사용한다.

NewHope NewHope [4]는 1라운드에 제출되었던 많은 RLWE 기반 알고리즘 중 2라운드에 진출한 경쟁력 있는 알고리즘으로, 대역폭 소모와 성능 사이에 좋은 절충

점을 갖고 있다. 중심이항분포를 사용한 에러샘플링에 대한 안전성을 처음 증명하였으며, 다항식 고속 곱셈알고리즘인 Number Theoretic Transform (NTT)를 처음으로 사용하였다. NTT 사용을 위해 모듈러 값을 고정하는 등 파라미터 설정이 까다로운 단점이 있어서 다양한 보안 수준을 지원하지 못한다.

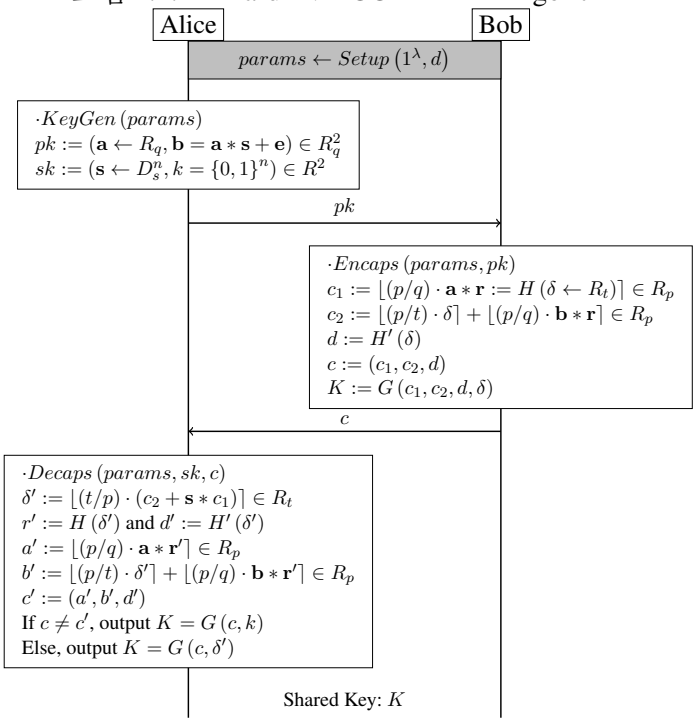
Round5 Round5 [2]는 후보 알고리즘 중 유일하게 Ring구조(RLWR)와 표준 구조(LWR)를 파라미터 조정만으로 선택할 수 있다. 두 가지 옵션을 응용 환경에 따라 선택할 수 있는 것은 큰 장점이다. 제 3절에서도 언급한 바와 같이 RLWR에서는 독특한 $\mathbb{Z}_q[X]/(X^{n+1}-1)$ 다항식(n+1은 소수)을 사용한다. 덕분에 을 요구하는 보안성 대비 과도하게 설정하지 않을 수 있어서, 대역폭 소모를 크게 줄였다. 다항식의 모듈러 연산이 어려운 단점은 곱셈 전 (X-1)을 곱하여 NTRU 다항식(X^n-1)으로 변환하여 연산을 하고 난 뒤 다시 (X-1)을 나누는 lift/unlift 방법을 사용하였다. 또한, 오류정정 부호화 기법인 XE5를 사용하여 5bit 오류를 정정함으로써 복호화 실패확률을 보상하고 대역폭을 줄였다. 그 결과 후보알고리즘 중 가장 적은 대역폭 소모면서도 경쟁할 만한 속도를 보인다.

LAC LAC [7]은 RLWE에 기반하며 기본 설계 개념은 NewHope와 유사하다. 다만, 대역폭 절약을 위해 다항식 계수의 모듈러스를 251로 작게 설정하고, 이로 인해 증가하는 복호화 실패확률은 오류정정 부호화 기법인 BCH를 사용하여 보정하였다. 그 결과 NewHope에 비해 대역폭이 적으면서도 대등한 성능을 보인다.

ThreeBears ThreeBears [18]는 Integer-MLWE라고 불리는 독특한 문제에 기반으로 한다. 이 문제는 다른 기반 문제에 비해 잘 연구되지 않은 단점이 있으나, 기존에 최적화되어 있는 정수 연산 라이브러리를 사용할 수 있어 쉽게 고속 구현이 가능한 장점이 있다. LAC이나 Round5과 같이 오류정정 부호화 기법을 사용하고, 고속 곱셈 기법으로 Karatsuba 알고리즘을 사용한다. NIST는 Integer-MLWE를 활용해 설계한 알고리즘이라는 독창성에 흥미로움을 표현했지만, 세부적인 분석을 요구하고 있다.

제 5 절 RLizard

그림 2.1: RLizard IND-CCA2 KEM Algorithm



RLizard는 NIST 표준화 공모 1라운드에 진출한 Lizard의 Ring 변환 버전의 키 교환 매커니즘 및 공개키 암호 알고리즘으로, 동작과정은 그림 2.1과 같다. RLizard는 키 생성에는 RLWE문제를 사용하고, 암복호화에는 RLWR을 사용했다. RLizard를 위에 언급한 일반적인 변형 관점에서 살펴보면, 매우 보수적 안전성을 갖으면서도 실용적 측면을 많이 고려했음을 알 수 있다. 먼저, 키 생성에 필요한 에러샘플링을 이산 가우시안분포에서 누적 분포 표(Cumulative Distribution Table, CDT) 샘플링 기법을 사용하여 본래 RLWE 문제를 크게 변형하지 않는 고정밀도 샘플러를 사용한다. 암복호화의 고속 연산과 정확성 향상을 위해 s는 희박한 삼진 값(sparse trinary)을 사용하였다. 또한 s에는 해밍 웨이트 값을 부여함으로써 최악의 경우 s의

해밍 웨이트가 매우 낮아져 공격에 노출될 수 있는 위험을 줄였다. 또한, IND-CCA2 안전성을 갖는 PKE와 KEM을 만들기 위해 IND-CPA에서 안전성이 증명된 PKE를 바탕으로 Fujisaki-Okamoto transformation [19]을 사용하였다. RLizard의 장점은 첫째, (R)Lizard는 효율적인 암복호화 연산을 수행한다. 또한, 에너지 소비량도 후보 알고리즘 중 가장 적은 수준으로 평가 [20]되어 저전력을 요구하는 IoT에도 적합한 것으로 볼 수 있다. 둘째, 강력한 보안 수준을 제공한다. RLizard는 파라미터 설정에 따라 최대 300bit 이상의 보안 강도를 가질 수 있다. 이는 NIST 표준화 후보알고리즘들과 비교하여도 가장 강력한 보안 강도이다. 또한, negligible한 복호화 실패확률을 갖고 있어 정확성이 낮은 알고리즘들을 공격하는 특수한 방법으로부터 안전한다. 하지만, 다른 알고리즘에 비해 공개키와 암호문 크기가 매우 크다. 특히, 대역폭과 성능이 중요한 척도인 Ring구조에서 NewHope 대비 4배 이상 대역폭이 크다. 이것은 군에서 지향하고 있는 IoT, 드론 등 경량화 장비의 암호 적용에 큰 걸림돌로 작용할 수 있다. 또한, 에러 샘플링을 위해 가우시안 분포에서 CDT 방법을 사용하는데, 표 검색에 사용되는 시간이 일정하지 않는 등의 문제로 인해 캐시 및 타이밍 공격과 같은 부채널 공격에 노출되어 있다.

제 3 장 국방 환경에 적합한 양자내성 암호

제 2장 3절에서 5절까지 살펴본 바와 같이 RLWE 기반 알고리즘은 안전성, 정확성, 대역폭 및 성능을 향상시키기 위해 다양한 기법을 적용하며 발전해왔다. 하지만, NIST 표준화 후보알고리즘 및 RLizard에서 알수 있듯 모든 것을 측면을 만족시키는 알고리즘을 하나만 선택하기는 어렵다. 또한 국방 환경에서 중요한 척도인 부채널 공격에 대한 내성은 거의 반영되지 않은 상태이며, 안전성과 정확성에 영향을 주는 에러 종속성 영향 등 최신 연구를 반영할 경우 탁월한 하나의 알고리즘을 고르는 일은 거의 불가능 하다. 따라서, 이 연구에서는 안전성과 정확성, 대역폭, 성능, 부채널 저항성 등 모든 면에서 우수한 알고리즘을 만들기 위해 NIST 후보 및 RLizard의 장점을 조합하고 최신 연구 결과를 병합하였다. 그 결과물로 양자내성을 갖는 키 교환 메커니즘(Key encapsulation mechanism, KEM)과 공개 키 암호 알고리즘(Public-key encryption)인 LizarMong을 설계하였다. LizarMong은 NIST 공모 1라운드에 진출한 Lizard[1]의 Ring 변환 버전인 RLizard를 바탕으로, 2라운드 후보 알고리즘들이 사용한 오류 정정 부호화 기법 및 암호문/공개 키 압축 기법을 적용하고, 키 생성에 사용되는 에러샘플러를 변경하여 대역폭과 연산 속도의 향상을 도모하였다. 또한, 최근 연구인 다양한 부채널 공격과 bit 간 종속성을 고려한 정확성 계산을 통해 안전성과 정확성을 강화하였다. LizarMong은 128bit 보안강도인 `Comfort`와 256bit 보안강도인 `Strong` 버전으로 사용할수 있으며, 공격자가 평문을 선택할 수 있는 상황 하에서의 구별 공격(Indistinguishability under chosen-plaintext attack, IND-CPA)에 안전한 공개 키 암호 알고리즘(Public-key Encryption, PKE)와 가장 강력한 공격자 가정인 적응형 선택 암호문 상황하 구별 공격(Indistinguishability under adaptive chosen ciphertext attack, IND-CCA2)에도 안전한 키 교환 매커니즘(Key Encapsulation Mechanism, KEM)을 지원한다.

이번 장에서는 국방 환경에 적합한 양자내성 암호인 LizarMong에 대해 세부적

으로 기술한다. 1절에서는 알고리즘 설계 요소의 선택과 이유를 기술하였다. 2절에서는 알고리즘을 상세히 설명하고, 3절에서는 각 보안강도를 만족시키기 위한 파라미터 설정과 해당 파라미터 별 대역폭을 기술하였다. 제 4절은 안전성을 증명하고 알려진 공격에 대한 해독 시간복잡도를 추정하였다. 5절에서는 정확성을 평가하였고 6절에서는 알려진 부채널공격에 대해 내성을 갖기 위한 방안을 제시하였다.

제 1 절 설계 요소 선택

Ring 선택 정수 환 $R_q := \mathbb{Z}_q[X]/(f(X))$에서 $f(X)$를 n이 2의 지수승인 특별한 형태의 cyclotomic 다항식 $X^n + 1$로 사용한다. 이 형태의 Ring은 RLizard를 비롯하여 NIST의 2라운드 후보 알고리즘 대부분이 사용하는 일반적인 형태이다. 이 형태의 Ring은 다항식 모듈러스 연산이 매우 간단하고 동형암호 등에서도 사용되면서 가장 잘 연구되었으며, 대수적 구조를 악용한 특별한 공격이 알려진 바 없어 안전성 측면에서도 가장 보수적 접근이다 [21].

모듈러 선택 LizarMong은 대역폭을 절감하고 성능을 높이기 위해 모듈러스 q를 2의 지수승 형태로 작게 설정한다. RLWE는 모듈러스 q의 값과 관계없이 어렵다는 것이 증명[22] 되었기 때문에, 이 선택은 RLWE의 어려움을 해치지 않는다. LizarMong은 특별히 q를 256으로 고정한다. 이는 컴퓨터 연산의 기본인 1byte 크기로 모듈러스 연산을 오버헤드 없이 수행할 수 있으며 메모리 사용 측면에서도 매우 효율적이다. 특히, AVX2 등 SIMD 연산을 지원하는 프로세서에서는 더욱 유리하게 작용한다. RLWR에 사용되는 모듈러스 p와 k도 2의 지수승으로 사용한다. 이를 통해 $\lfloor (p/q) \cdot \mathbf{x} \rceil$ 연산을 bit 간 덧셈과 AND 연산으로 대체 [23]함으로써 성능을 향상시킬 뿐만 아니라, 모듈러스 연산을 악용한 timing attack을 불가능하게 한다 [24].

분포 선택 모든 NIST 2라운드 후보 알고리즘들은 모든 RLWE 계열 2라운드 후보 알고리즘들은 효율적인 구현을 위해 비밀 $\mathbf{s}$와, 에러 $\mathbf{e}$를 같은 분포에서 $seed$ 값만 다

르게 넣어 샘플링한다. 이 방법은 안전성이 증명 [25]되어 그 동안 알고리즘 설계의 표준처럼 이용되었다. 하지만, 최근 Fault attack으로 샘플러에 같은 $Seed$를 주입하여 $\mathbf{s}$와 $\mathbf{e}$를 동일하게 만들어 해독을 시도하는 공격이 발표되었다 [26]. 그러므로, $\mathbf{s}$와 $\mathbf{e}$를 동일 분포에서 샘플링 하기 위해서는 별도의 대응기법이 요구된다. 따라서, 이 연구에서는 Fault attack 공격포인트를 제거하기 위해 본래의 RLWE에서와 같이 에러와 비밀 값을 각각 다른 분포에서 샘플링한다.

- **에러 분포** LizarMong은 에러 다항식 $\mathbf{e}$를 중심이항분포에서 샘플링한다. 이는 이산 가우시안 분포 구현의 오버헤드를 제거하기 위한 방법으로, 빠른 속도를 가지면서 PKE/KEM 환경에서 안전성이 증명 [4]되었다는 장점이 있다. 또한, RLWE에 알려진 공격은 오류 샘플링 분포의 종류와 무관하게 표준편차에만 의존하므로 더욱이 안전성에는 문제가 없다. 참고로, RLizard는 이산 가우시안 분포를 바탕으로 CDT 샘플링 기법을 사용하였다. CDT 샘플링은 고정밀도 샘플링이 가능하고 빠른 장점이 있지만 PKE/KEM 환경에서는 높은 정밀도를 필요로 하지않으며 테이블 검색을 기반으로 하여 cache 공격[27]에 약점을 갖고 있다.

- **비밀 분포** 본래 RLWE 문제는 비밀 다항식 $\mathbf{s}$를 R_q에서 유니폼 랜덤하게 선택한다. 하지만, 큰 비밀 값을 곱하게 되면 오류도 커져서 알고리즘의 정확성을 떨어뜨리고 연산량이 많아지는 문제가 있다. [23]과 [2]는 각각 LWE와 LWR에서 비밀 값을 이진 또는 삼진에서 0이 아닌 값이 희박하게(sparse) 샘플링하는 경우도 안전성이 동일하다는 것을 증명하였다. 이 증명에 따라 RLizard, LAC, Round5 등은 희박한 삼진 분포에 해밍 웨이트를 적용한 비밀 값 샘플링 분포 $HWT_n(h)$를 사용한다. 이 분포를 사용하면 정확성이 높아지고 다항식 곱셈을 덧셈과 뺄셈의 형태로 바꾸어 고속 곱셈 연산도 가능[28]한 장점이 있다.

오류정정 부호화 기법 오류정정 부호화 기법은 [14]에서 분석한 바와 같이 적은 대역폭을 유지하면서 안전성과 정확도를 높이기 위한 유용한 툴로 RLWE 계열에 사용

될 수 있다. 2라운드 후보 알고리즘 중 LAC, ThreeBears, Round5가 사용하고 있다. LizarMong은 에러를 발생시키거나 확대시키는 주요 원인인 $\mathbf{s}$와 $\mathbf{e}$를 희박한 삼진 분포에서 샘플링하기 때문에 상대적으로 복호화 실패 확률이 낮다. 하지만, 에러비율을 결정하는 모듈러스 크기 또한 작기 때문에 오류 정정 부호화 기법 없이 사용하는 것은 곤란하다. LAC은 8bit 이상 많은 오류 정정을 위해 성능 오버헤드를 감수하고도 BCH 오류 정정 부호화 기법[29]을 사용한다. 특히, BCH는 인코딩에 비해 디코딩 시 성능 오버헤드가 큰데, RLWE 계열이 암호화에 비해 복호화가 느리다는 점을 감안할 때 디코딩 시의 오버헤드는 치명적이다. Threebears는 2bit의 에러만 정정하면 되고, 상대적으로 작은 차원으로 설계되어 패리티 bit가 적고 상수 시간 구현이 손쉬운 Melas code를 사용했다. Round5의 경우 [30]에서 설계한 오류 정정 부호화 기법 XE5를 사용한다. XE5는 패리티 bit가 거의 메시지 길이만큼 크지만(256bit 메시지에 234bit 패리티 bit) 인코딩과 디코딩이 매우 빠르고, 상수 시간 구현이 쉽고 분기점이 없어 부채널 공격에 강한 장점이 있다. LizarMong은 4.3절 복호화 실패 확률의 분석에 따라 4-5bit의 오류 정정 능력이 필요하기 때문에 XE5를 사용한다. XE5는 메시지 길이가 256bit로 고정되어 있으므로, Strong 파라미터의 512bit 메시지를 입력 값으로 받을수 없다. 그래서 Strong에서는 메시지를 반으로 나누어 각각 XE5를 취하여 코드워드를 만든 뒤 이를 병합하는 간단한 트릭을 사용한다. 이는 명백히 안전성에 영향을 주지 않는다.

공개 키 압축 기법 RLWE 기반 공개 키 암호에서 공개 키는 R_q에서 유니폼 랜덤하게 선정한 다항식 $\mathbf{a}$와 비밀 키 $\mathbf{s}$를 곱하고 오류 $\mathbf{e}$를 더한 $\mathbf{b}$로 구성된다. 공개 키 $\mathbf{a}, \mathbf{b}$는 각각 압축 가능하다. $\mathbf{a}$의 압축방법은 $\mathbf{a}$를 모두 전송하는 대신 $seed$만 보내고, 동일한 해시함수(`SHAKE256` 등)로 복구하는 방법이다. 이 방법은 공개 키 크기를 $2 \times (n \log q)$에서 $size - of - seed + (n \log q)$로 감소시킨다. $\mathbf{b}$의 압축 방법은 모듈러스를 q보다 작은 값인 k로 줄이는 방법으로, 일부 하위 bit를 버리고 전송하는 개념이다. 이는 RLWR의 아이디어와 흡사하다. $\mathbf{b}$의 크기를 $(n \log q)$에서 $(n \log k)$

로 줄일수 있다. NIST 공모 2라운드의 모든 알고리즘이 $\mathbf{a}$ 대신 $seed$만 보내는 압축방식을 택하고 있는 반면, $\mathbf{b}$ 압축은 NewHope과 1라운드 버전의 KYBER에서만 사용하는데, 이는 안전성에 영향을 주기 때문이다 [7]. 실제로 KYBER는 2라운드에서 $\mathbf{b}$ 압축을 제거하였다. 따라서, LizarMong은 잘 증명된 RLizard의 안전성에 손상을 가하지 않기 위해 $\mathbf{a}$의 압축 방법만을 사용하며, 이때 $seed$의 크기는 256bit로 선택하였다.

암호문 압축 기법 LizarMong은 NIST 공모 2라운드에 진출한 모든 알고리즘에서 사용한 암호문 압축 기법을 사용한다. RLizard의 경우 RLWR 기반의 암복호화를 통해 암호문을 압축하는 효과를 얻어 추가적인 압축을 하지 않았다. 하지만, 암복호화를 RLWR 기반으로 수행하는 SABER와 Round5도 대역폭을 줄이기 위해 암호문을 추가적으로 압축한다. 따라서, LizarMong도 추가적인 암호문 압축을 수행한다. c_1, c_2를 전부 압축하는 경우[5]도 있으나, c_1을 압축할 경우 안전성에 영향을 미치는 것으로 알려져있다[7]. 따라서, LizarMong은 NewHope, Round5, SABER 등 대부분의 알고리즘에서 선택한 c_2만 압축하는 방법을 사용한다. 참고로 RLizard 경량화 버전[21]에서는 LizarMong과 같이 공개 키 및 암호문 일부 압축 기법을 사용한다.

제 2 절 상세 알고리즘

1. IND-CPA PKE

Algorithm 1 IND-CPA.KeyGen

Input: The set of public $parameters$

Output: Public key $pk = (Seed_a \parallel \mathbf{b})$, Private Key $sk = (\mathbf{s})$

1: $Seed_a \xleftarrow{\$} \{0,1\}^{256}$

2: $\mathbf{a} \leftarrow \texttt{SHAKE256}(Seed_a, n/8)$

3: $\mathbf{s} \xleftarrow{\$} HWT_n(h_s)$ and $\mathbf{e} \xleftarrow{\$} \psi_{cb}^n$

4: $\mathbf{b} \leftarrow -\mathbf{a} * \mathbf{s} + \mathbf{e}$

5: $pk \leftarrow (Seed_a \parallel \mathbf{b})$ and $sk \leftarrow \mathbf{s}$

6: **return** pk, sk

Algorithm 2 IND-CPA.Encryption

Input: pk, Message $\mathbf{M} \in \{0,1\}^d$

Output: Ciphertext $\mathbf{c} = (\mathbf{c_1} \parallel \mathbf{c_2})$

1: $\mathbf{r} \xleftarrow{\$} HWT_n(h_r)$ and $\mathbf{M}' \leftarrow$ eccENC($\mathbf{M}$)

2: $Seed_a, \mathbf{b} \leftarrow$ Parsing(pk)

3: $\mathbf{a} \leftarrow \texttt{SHAKE256}(Seed_a, n/8)$

4: $\mathbf{c_1} \leftarrow \lfloor (p/q) \cdot \mathbf{a} * \mathbf{r} \rceil$ and $\mathbf{c_2} \leftarrow \lfloor (k/q) \cdot ((q/2) \cdot \mathbf{M}' + \mathbf{b} * \mathbf{r}) \rceil$

5: $\mathbf{c} \leftarrow (\mathbf{c_1} \parallel \mathbf{c_2})$

6: **return** $\mathbf{c}$

Algorithm 3 IND-CPA.Decryption

Input: sk, Ciphertext $\mathbf{c} = (\mathbf{c_1} \parallel \mathbf{c_2})$

Output: Message $\hat{\mathbf{M}}$

1: $\mathbf{c_1}, \mathbf{c_2} \leftarrow$ Parsing($\mathbf{c}$)

2: $\hat{\mathbf{M}}' \leftarrow \lfloor (2/p) \cdot ((p/k) \cdot \mathbf{c_2} + \mathbf{c_1} * \mathbf{s}) \rceil$

3: **return** $\hat{\mathbf{M}} \leftarrow$ eccDEC($\hat{\mathbf{M}}'$)

2. IND-CCA2 KEM

LizarMong은 IND-CCA2 KEM으로의 변환을 위해 Jiang이 최근 발표한 변환 기법 [31]을 사용한다. 이 기법은 기존에 RLizard가 사용했던 방법 [19]보다 안전성이 더 잘 증명되어 있으며, 추가적인 해시 함수 연산이 불필요하여 대역폭 절감에도 유리 하다. Jiang의 변환기법에 사용되는 해시함수는 $H : R_2 \rightarrow HWT_n(h)$, 그리고 $G : \{0,1\}^* \rightarrow \{0,1\}^n$ 로 정의 한다.

Algorithm 4 IND-CCA2-KEM.KeyGen

Input: The set of public $parameters$

Output: Public Key $pk = (Seed_a \parallel \mathbf{b})$, Private Key $sk = (sk_{cpa} \parallel \mathbf{u})$

1: $pk, sk_{cpa} :=$ IND-CPA.KeyGen (Algorithm 1)

2: $\mathbf{u} \xleftarrow{\$} R_2$

3: **return** $pk, sk \leftarrow (sk_{cpa} \parallel \mathbf{u})$

Algorithm 5 IND-CCA2-KEM.Encapsulation

Input: pk

Output: Ciphertext $\mathbf{c} = (\mathbf{c_1} \parallel \mathbf{c_2})$, Shared Key $\mathbf{K}$

1: $\delta \xleftarrow{\$} \{0,1\}^{sd}$

2: $\mathbf{r} \leftarrow H(\delta)$

3: $\delta' \leftarrow$ eccENC(δ)

4: $\mathbf{c_1} \leftarrow \lfloor (p/q) \cdot \mathbf{a} * \mathbf{r} \rceil$

5: $\mathbf{c_2} \leftarrow \lfloor (k/q) \cdot ((q/2) \cdot \delta' + \mathbf{b} * \mathbf{r}) \rceil$

6: $\mathbf{c} \leftarrow (\mathbf{c_1} \parallel \mathbf{c_2})$

7: $\mathbf{K} \leftarrow G(\mathbf{c}, \delta')$

8: **return** $\mathbf{c}, \mathbf{K}$

Algorithm 6 IND-CCA2-KEM.Decapsulation

Input: pk, sk, Ciphertext $\mathbf{c}$

Output: Shared Key $\mathbf{K}$

1: $\mathbf{c_1}, \mathbf{c_2} \leftarrow \text{Parsing}(\mathbf{c})$

2: $sk_{cpa}, \mathbf{u} \leftarrow \text{Parsing}(sk)$

3: $\hat{\delta}' \leftarrow \lfloor (2/p) \cdot ((p/k) \cdot \mathbf{c_2} + \mathbf{c_1} * sk_{cpa}) \rceil$

4: $\hat{\delta} \leftarrow \text{eccDEC}(\hat{\delta}')$

5: $\hat{\mathbf{r}} \leftarrow H(\hat{\delta})$

6: $\hat{\delta}'' \leftarrow \text{eccENC}(\hat{\delta})$

7: $\hat{\mathbf{c}} \leftarrow \lfloor (p/q) \cdot \mathbf{a} * \hat{\mathbf{r}} \rceil \parallel \lfloor (k/q) \cdot ((q/2) \cdot \hat{\delta}'' + \mathbf{b} * \hat{\mathbf{r}}) \rceil$

8: **if** $\mathbf{c} \neq \hat{\mathbf{c}}$ **then** $\mathbf{K} \leftarrow G(\mathbf{c}, \mathbf{u})$ **else** $\mathbf{K} \leftarrow G(\mathbf{c}, \hat{\delta}'')$

9: **return** $\mathbf{K}$

제 3 절 파라미터

파라미터는 NIST 양자내성암호 표준화 공모[32]에서 요구한 Category1 (128bit)을 만족시키는 `Comfort`와 Category5(256bit)를 만족시키는 `Strong`으로 선정한다. 보안 수준에 대한 평가는 4.2절에 기술한 알려진 모든 공격에 대한 계산복잡도를 반영하였다. 참고로, [32]에서는 192bit 보안수준인 category3도 요구하고 있지만 LizarMong의 경우 `Strong` 파라미터로도 다른 알고리즘의 192bit 보안수준과 대역폭 및 성능이 대등하기 때문에 고려하지 않았다. 보안수준별 세부 파라미터는 표 3.1에 정리하였다. n은 격자의 차원(dimension), q는 LWE의 모듈러스, p는 LWR의 모듈러스, k는 암호문 압축에 사용되는 모듈러스, h_s는 비밀 키의 해밍 웨이트, h_r은 암호문 생성이 필요한 임시 비밀 값의 해밍 웨이트이다. d는 메시지의 길이이고, sd는 IND-CCA2 변환 시 사용되는 δ의 길이이다. cb는 중심이항분포에 사용되는 변수이다. 파라미터에 따른 대역폭은 표 3.2에 정리하였다. 키 교환 메커니즘에서

Alice (server)는 공개 키를 Bob(client)에게 전송하고, Bob은 공개 키를 활용해 임시 비밀 값(δ)를 암호화 하여 Alice에게 재전송하고 공유 비밀키를 갖는다. Alice는 Bob이 보낸 암호문을 비밀 키로 해독하여 δ를 얻어 공유 비밀키를 생성한다. 이러한 연산과정을 거치기 때문에 대역폭은 Alice가 Bob에게 보내는 공개 키의 크기, Bob이 Alice에게 보내는 암호문의 크기로 정의된다.

표 3.1: 보안 강도에 따른 세부 파라미터

$parameters$	n	q	p	k	h_s	h_r	d	sd	cb
Comfort	512	256	64	16	128	128	256	256	1
Strong	1024	256	64	16	128	128	512	512	1

표 3.2: 공개키, 비밀키 및 암호문의 크기(단위: bytes)

보안 수준	**암호문**	**공개키**	**비밀키**
`Comfort`	640	544	544
`Strong`	1280	1056	1088

제 4 절 안전성 분석

이 연구에서의 안전성 분석은 Ring-LWE에 대한 [4]의 매우 비관적인 평가와 [23]의 sparse trinary 비밀을 갖는 Ring-LWE 및 Ring-LWR에 대한 안전성 증명, [33]의 공격량(복잡도) 계산 방식을 바탕으로 한다.

1. IND-CPA 및 IND-CCA2 증명

RLizard.CPA가 IND-CPA에서 안전하다는 가정하에 LizarMong의 IND-CPA 안전성을 증명한다.

Lemma 1 *RLizard는 주어진 파라미터에 대한 Ring-LWE, Ring-LWR문제의 어려움을 가정할 때, IND-CPA secure하다[1].*

Algorithm 7 KeyGen'

Input: The set of public parameters
Output: Public key $pk' = (\mathbf{a}, \mathbf{b})$

1: $pk = (Seed_a, \mathbf{b}) \leftarrow LizarMong.KeyGen$
2: $\mathbf{a} \leftarrow \texttt{SHAKE256}(Seed_a, n)$
3: $pk' \leftarrow (\mathbf{a}, \mathbf{b})$
4: **return** pk'

Theorem 1 *LizarMong은 주어진 파라미터에 대한 Ring-LWE, Ring-LWR문제의 어려움과* `SHAKE`*가 Random Oracle이라고 가정할 때, IND-CPA secure하다.*

$LizarMong$의 Homomorphic 성질에 의해 메시지 $m = 0$에 대한 안전성만 증명하면 충분하다. $LizarMong'$은 LizarMong에서 암호문 압축을 하지 않는 알고리즘이라고 정의하자. 먼저 $LizarMong'$가 IND-CPA secure 함을 보인다. $KeyGen'$을 Algorithm 7 같이 정의하자.
분포 D_0, D_1, D_2를 다음과 같이 정의하자.

$$D_0 = \{(pk', C) : pk' \leftarrow KeyGen'(params), C = (c_1, c_2) \leftarrow LizarMong.Enc_p k(0)\}$$
$$D_1 = \{(pk, C) : pk \leftarrow RLizard.KeyGen(params), C = (c_1, c_2) \leftarrow RLizard.Enc_p k(ecc(0))\}$$
$$D_2 = \{(pk, C) : pk \leftarrow Ring, C = (c_1, c_2) \leftarrow Ring\}$$

`SHAKE`는 Random Oracle이므로, pk'와 pk의 분포는 구분 불가능하다. 또한 $RLizard$와 $LizarMong$의 Encryption 함수 정의에 따라 $pk = (Seed_a, b)$에 대한 $LizarMong$의 암호문 $C_{LizarMong} \leftarrow LizarMong.Enc_{pk}(0)$과 $pk' = (\texttt{SHAKE256}(Seed_a, n), b)$에 대한 $RLizard$의 암호문 $C_{RLizard} \leftarrow RLizard.Enc_{pk'}(ecc(0))$은 동일하다. 즉, $C_{LizarMong} = C_{RLizard}$이다. 따라서 D_0과 D_1은 계산적으로 구분 불가능하다. 또한 Lemma 1에 의해 D_1과 D_2는 계산적으로 구분 불가능하다. 따라서 D_0와 D_2는 계산적으로 구분 불가능하다. 즉, $LizarMong'$은 IND-CPA secure하다. [7]에 따라 암호문 압축은 LWE, LWR 기반 공개 키 암호의 안전성에 영향을 주지 않으므로,

$LizarMong$은 IND-CPA secure 함을 알 수 있다. 또한, LizarMong은 Theorem 1에 따라 IND-CPA에서 안전한 PKE이며, 4.3절의 복호화 실패 확률 계산에 따라 negligible한 복호화 실패확률을 갖고 있기 때문에 Jiang의 변환기법 [31]을 활용해 IND-CCA2 KEM으로 변환해도 안전하다.

2. 알려진 공격에 대한 안전성 분석

알려진 공격 방법에 대한 계산복잡도를 평가하여 안전성을 측정한다. 이러한 안전성 분석 방법은 모든 RLWE 계열 알고리즘에서 적용하는 방법이다. 이 연구에서 고려한 RLWE에 대한 공격은 [33]의 방법들이며, 특별히 희박한 삼진 비밀 값(Sparse trinary secret)에 대한 공격을 고려하기 위해 [34]도 검토하였다. RLWE 공격에 대한 계산복잡도 평가하는 범용적인 도구인 online LWE estimator[35]을 사용하여 알려진 모든 공격에 대한 평가를 수행한 결과, BKZ 격자 기저 감소 알고리즘을 사용하는 [34] primal 공격이 가장 위협적이다. BKZ 알고리즘은 격자의 차원을 감소시켜 작은 격자에 대해서 다항식 수준의 SVP oracle을 반복하여 수행해서 해결하는데, 이 반복횟수에 대한 측정에는 여러 논의가 있다. 이 연구에서는 방어자 입장에서의 매우 비관적인 분석 방법[4]으로 다항식 수준의 반복 횟수를 무시하고 한 번의 oracle 호출로 평가하는 core SVP 방법을 적용한다. 이에 따른 BKZ 격자 기저 감소 알고리즘의 계산 복잡도 2^{cn}이다. 이때 c는 constant값으로 Classical 환경일 때 $c = 0.292$, Quantum 환경에서는 $c = 0.265$이다. LWR 공격에 대한 공격복잡도는 동일한 차원과 q값, 그리고 오류 비율이 $p^{-1}\sqrt{\pi/6}$인 LWE 공격의 계산복잡도와 동일하다[23]. 이에 따라 LWR 공격에 대한 계산복잡도는 LWE 공격에 대한 계산복잡도와 유사하게 평가하였다.

[35]의 estimator를 통해 LWE, LWR에 대한 [34] 공격을 적용했을 때 계산복잡도는 표 3.3에 정리하였다. online estimator에 사용한 코드는 https://github.com/LizarMong 에서 확인할 수 있다. 결론적으로 LizarMong `Comfort`의 경우 128bit 보안 강도를 만족하고, `Strong`은 256bit 보안 강도를 만족한다.

표 3.3: 파라미타 별 복잡도 평가 결과

Parameters	Claim Security	Attacks		Classical	Quantum
Comfort	NIST Category 1 (AES 128bit)	Primal	LWE	**133**	**121**
			LWR	144	131
		Dual	LWE	165	154
			LWR	180	170
Strong	NIST Category 5 (AES 256bit)	Primal	LWE	**256**	**236**
			LWR	269	249
		Dual	LWE	304	275
			LWR	328	301

제 5 절 정확성(복호화 실패 확률)

복호화 실패는 오류를 포함함으로써 안전성을 담보받는 RLWE 계열 알고리즘에는 피할 수 없는 위험이다. 복호화 실패 확률은 알고리즘의 정확성을 의미함은 물론이고, 실패 확률을 악용한 공격[36, 37]으로 인해 안전성에도 영향을 줄 수 있는 중요한 요인이다. 지금까지 설계된 RLWE 계열 알고리즘의 실패 확률 계산은 각 bit의 오류가 독립적으로 발생한다는 가정을 두고 분석하였다. 하지만 최근 [8]는 각 bit 간의 오류가 독립적으로 발생하는 것이 아니라는 반증을 이론 및 실험적으로 밝혔다. [8]에 따르면, 각 bit 간 오류 발생 확률이 독립이 아니더라도, 오류 정정 부호화 기법을 사용하지 않는 경우에는 독립 가정을 바탕으로 한 계산이 유효하지만, 오류 정정 부호화 기법을 사용하는 경우는 부적절하다고 주장한다. LizarMong은 오류 정정 부호화 기법을 사용하기 때문에 각 bit의 오류가 종속적으로 발생한다는 가정하에 실패 확률 계산을 수행한다. RLizard는 $|\mathbf{e} * \mathbf{r} + \mathbf{s} * \mathbf{f}| \geq \frac{q}{4} - \frac{q}{2p}$를 만족하는 경우 복호화가 실패하게 된다 [1]. 이때, $\mathbf{f} = \mathbf{a} * \mathbf{r} - (q/p) \cdot \mathbf{c_1}$이다. LizarMong의 경우 암호문 압축에 따른 에러가 더 발생하게 된다. 따라서 LizarMong의 복호화 오류는 $\mathbf{v} := \lfloor (p/q) \cdot ((q/2) \cdot \mathbf{M}' + \mathbf{b} * \mathbf{r} \rceil$와 $\hat{\mathbf{v}} \leftarrow \mathbf{c_2} \ll (\log p - \log k)$의 차 만큼의 오류가 더 발생하게 된다. 따라서 $\mathbf{g} = \mathbf{v} - \hat{\mathbf{v}}$라고 정의하면 [8]에 따라 LizarMong

표 3.4: 복호화 실패 확률

Prameters	without ECC	with XE5(5bit ECC)
Comfort	2^{-37}	2^{-179}
Strong	2^{-68}	2^{-302}

은 $|\mathbf{e} * \mathbf{r} + \mathbf{s} * \mathbf{f} + \mathbf{g}| \geq \frac{q}{4} - \frac{q}{2p}$를 만족하는 경우 복호화가 실패하게 된다. 실패 확률을 계산하기 위해 $S = (\mathbf{s}, \mathbf{e})^T$, $C = (\mathbf{f}, \mathbf{r})^T$를 정의한다. 각 bit의 오류가 종속적으로 발생한다는 가정하에 복호화 실패 확률은 식 (3.1)에 따라 계산된다[8]. 이때, $\Pr[Fail]$은 복호화가 실패할 확률, $\Pr[F_i]$는 i번째 비트에서 오류가 발생할 확률, $Binom(k, n, p) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$, $p_b = \Pr[F_0 \mid \|S\|, \|C\|]$이다.

$$\Pr[Fail] \approx \sum_{\|S\|, \|C\|} (1 - Binom(d, l_m, p_b)) \cdot \Pr[\|S\|] \cdot \Pr[\|C\|] \tag{3.1}$$

또한 $p_b = \Pr[F_0 \mid \|S\|, \|C\|]$는 식 (3.2)에 따라 계산된다[36].

$$\begin{aligned} p_b = \sum_{l} \sum_{g_0} (& \Pr[|C^T S + \mathbf{g}|_0 > q/4 - q/2p \mid |C^T Ss|_0 = l, \mathbf{g}_0] \cdot \\ & \Pr[|C^T S|_0 = l \mid \|S\|, \|C\|] \cdot \Pr[\mathbf{g}_0]) \end{aligned} \tag{3.2}$$

위의 계산 과정을 파이썬 코드로 구현하였으며 https://github.com/LizarMong 에서 확인할 수 있다. 이러한 계산에 따르면, LizarMong의 복호화 실패 확률은 Comfort 파라메타에서 2^{-179}, Strong 파라메타에서 2^{-302}의 실패 확률을 갖는다.

제 6 절 부채널 공격

이 연구에서는 표 3.5에 열거한 대표적인 부채널 공격들을 고려하여 이 공격들로부터 저항성을 갖도록 알고리즘을 설계하였다. LizarMong의 대응전략의 우선순위는 각 공격들이 노린 취약점을 살펴보고 제 3장 1절의 설계 요소 선택 단계에서 가급적

해당 취약점을 제거하여 공격 표면을 최대한 줄이는 것이다. 그럼에도 불구하고 알고리즘 작동 상 제거할 수 없는 취약점들은 효율적인 대응기법을 찾아 알고리즘에 내재화 시키는 전략을 수립했다.

첫번째 대응전략에 따라, LizarMong은 알려진 Cache attack과 Timing attack에 안전하다. 간략히 설명하면 모듈러스 연산의 수행여부에 따른 시간 차이를 악용한 [24]의 timing attack은 모듈러스를 2의 지수승으로 사용하여 덧셈과 AND 연산으로 모듈러스 연산을 대체하는 LizarMong의 경우에 해당되지 않는다. 또한, LizarMong은 에러 샘플러를 중심이항분포로 사용함에 따라 RLizard가 사용한 CDT 샘플링을 악용한 [38]의 timing attack과 [27]의 cache attack에 안전하다. 한편, 설계 요소 선택의 추가적인 이점으로 일부 differential attack과 fault attack을 피할 수 있다. $\mathbf{s}$와 $\mathbf{e}$를 동일한 분포에서 샘플링하는 상황을 공격한 [26]의 fault attack은 $\mathbf{s}$와 $\mathbf{e}$를 다른 분포에서 샘플링하도록 설계한 LizarMong에는 적용되지 않는다. 추가적으로, 고속 곱셈연산 기법인 NTT를 공격하는 [39]의 differential attack은 NTT를 사용하지 않기 때문에 해당되지 않는다. 설계 요소 선택 단계에서의 부채널 공격 표면 최소화 노력에도 불구하고, 다항식 곱셈과 에러 생성 연산을 직접 노린 differential attack과 fault attack은 여전히 위협으로 남는다. 따라서, LizarMong에 남은 몇 가지 공격에 대한 방어대책을 추가하였다.

- **Differential Attacks** Differential Attack은 크게 두 집합 간의 차이를 이용하는 DPA (Differential Power Analysis)와 중간 값과 부채널 정보 간의 상관 계수를 이용하는 CPA (Correlation Power Analysis)로 구분된다. [40]은 Newhope를 대상으로 DPA가 가능함을 보였다. 이 공격은 공개 키와 비밀 키 간의 다항식 곱셈을 공격 포인트로 삼았다. RLWE 계열 알고리즘에서 다항식 곱셈은 반드시 필요하기 때문에 부득이 대응기법을 별도로 둘 수 밖에 없다. 알려진 대응기법으로는 마스킹(Masking) 방법[44, 45, 46]과 하이딩(Hiding) 방법[42]이 있다. 알려진 마스킹 방법은 랜덤 값을 더하는 디코더를 이용하여 구성하는 일반적인 방법과 특별히 RLWE 계열의 덧셈 동형 사상을 이용하여 복호화 연산을 반복 수행하는 방법이다. RLWE

표 3.5: 알려진 부채널 공격과 LizarMong의 대응기법

방 법	문 헌	공격 지점	LizarMong	방어기법
Timing	[24]	Modulus	X	-
	[38]	CDT sampling	X	-
Differential	[39]	INTT	X	-
	[40]	Multiplication	O	hiding
	[41]	Multiplication	O	hiding
Template	[42]	Multiplication	O	hiding
Fault	[43]	Error sampling	O	loop check
	[26]	Error and secret sampling	X	-
Cache	[27]	CDT sampling	X	-

계열의 덧셈 동형 사상을 이용하는 방법은 RLWE 계열만 사용할 수 있는 독창적 방법이나, RLWE 계열의 복호화 속도가 암호화 속도보다 느리다는 점에서 추가적인 성능 오버헤드는 부담된다. 하이딩은 곱셈 연산 순서를 뒤섞거나 더미 연산을 추가 하는 등의 방법이 있다. LizarMong은 differential attack으로부터 내성을 갖으면서도 연산 속도 오버헤드를 최소화하기 위해 [28]의 희박한 다항식 곱셈(Sparse polynomial multiplication) 알고리즘에 [42]의 하이딩 방법을 병합한 새로운 곱셈 방법인 Algorithm 8를 사용한다. 이 연구에서 설계한 하이딩 곱셈 방법은 연산 시작 인덱스를 랜덤하게 하는 방법으로, 랜덤 값을 1회만 추출해도 되므로 연산 순서를 뒤섞는 방법보다 오버헤드가 적다.

• **Fault Attacks** Fault attack은 직접 fault를 주입하는 공격 방식으로 공격자 가정이 강한 편이다. RLWE 계열에 알려진 공격 방법은 $\mathbf{s}$와 $\mathbf{e}$를 생성하는 과정을 공격노린다. $\mathbf{s}$와 $\mathbf{e}$를 생성할 때, 랜덤 값을 얻은 뒤 특정한 함수 또는 반복문을 거쳐 $\mathbf{s}$와 $\mathbf{e}$를 가공하여 생성하게 되는데, [43]는 fault 주입을 통해 반복문을 수행하지 못하게 하여 $\mathbf{s}$와 $\mathbf{e}$를 초깃값인 0으로 만든다. LizarMong은 $\mathbf{s}$와 $\mathbf{e}$를 랜덤 값 추출 후 반복문에서 가공하는 방식을 사용하여 [43] 공격에 노출되어 있다. 따라서, [47]

Algorithm 8 Sparse Polynomial Multiplication with Hiding Countermeasure

Input: $\mathbf{a} = \sum_{i=0}^{n-1} a(i) \cdot x^i \in R_q$, $\mathbf{r} = \sum_{i=0}^{n-1} r(i) \cdot x^i \in HWT_n(h)$,
$d = [i_0, \ldots, i_{g-1}, i_g, \ldots, i_{h-1}]$ with $d[k] = i_k$ such that $r(i_k) = 1$ for $k \in [0, g)$ and $r(i_k) = -1$ for $k \in [g, h)$

Output: $\mathbf{v} = \mathbf{a} * \mathbf{r} = \sum_{i=0}^{n-1} v(i) \cdot x^i \in R_q$

```
initialize v to zero polynomial                          ▷ size of v = 2n
m ←$ {0, 1, ..., h − 1}                                  ▷ random starting index
for i ∈ {0, ..., h − 1}, j ∈ {0, ..., n − 1} do
    if m + i (mod h) < g then
        v(d[m + i (mod h)] + j) = v(d[m + i (mod h)] + j) + a(j)
    else
        v(d[m + i (mod h)] + j) = v(d[m + i (mod h)] + j) − a(j)
    end if
end for
for i ∈ {0, ..., n − 1} do
    v(i) = v(i) − v(n + i)
end for
return v
```

에서 제안한 통계적 테스트를 수행함으로써 저항성을 갖도록 알고리즘에 반영하였다. LizarMong에서 사용한 통계적 테스트는 반복문에 사용된 인덱스의 최종 결과가 정확한지만 확인하는 절차를 거치면 되기 때문에 오버헤드가 적다.

제 4 장 평가

평가는 안전성(보안 수준), 정확성(복호화 실패 확률), 대역폭(암호문+공개키 크기), 성능(암복호화 연산속도) 측면에서 분석하였다. NIST 공모 2라운드 후보 알고리즘과 RLizard를 비교 대상으로 하였으며 각 비교 대상들의 안전성과 정확성, 대역폭은 NIST에 제출한 공식문서(RLizard는 1라운드 문서)를 참조했고, 성능(연산속도)는 NIST에 제출한 각 알고리즘의 Optimize 버전 code를 활용하였다. 공정한 성능평가를 위해 비교대상 알고리즘과 LizarMong을 동일한 환경에서 실행했으며, 평가 환경은 Intel i7-9700K@3.2GHz CPU, ubuntu 16.04.11, gcc 5.4.0 컴파일러에 옵션은 -O3로만 부여하여 1,000회 반복 후 평균 값을 구했다. 종합된 평가 결과는 표4.1과 그림4.1에 정리하였다. 표4.1에서 각 알고리즘별로 3개의 행은 128, 192, 256bit 보안 수준을 의미한다.(즉, LizarMong과 NewHope는 192bit 보안 수준을 지원하지 않는다.) 구현은 https://github.com/ LizarMong 에 코드를 공개하였다.

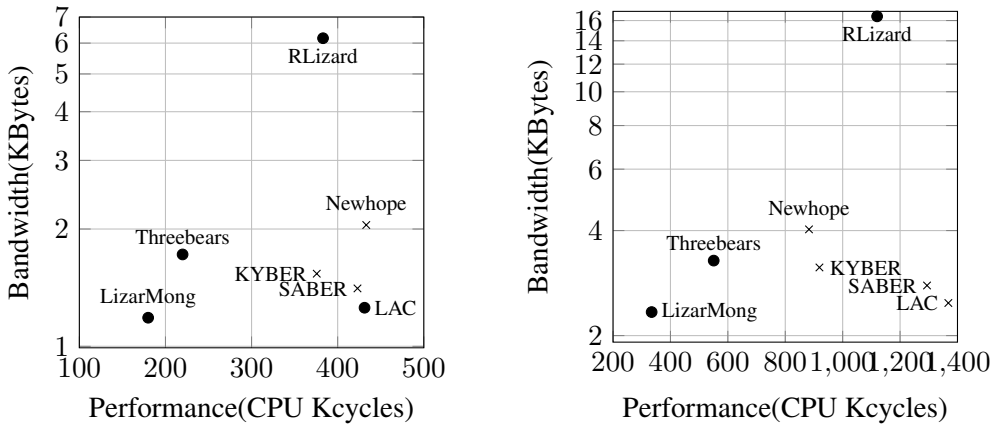


그림 4.1: KEM 기준 대역폭 및 성능 비교 (좌) 128bit 보안 강도 (우) 256bit 보안 강도 (참고: ● 는 IND-CCA2에서 안전성과 각 보안 수준과 유사한 정확성을 갖는 알고리즘이며, ×는 그렇지 못한 경우를 표시하였다.)

표 4.1: NIST 공모 2라운드 후보 알고리즘 및 RLizard의 KEM과의 평가 비교표

Algorithms	**Security** (log)	**Correctness** (log)	**Bandwidth** (Bytes)	**Performance** (K cycles)	
				Enc+Dec	**KeyGen**
LizarMong	133	−179	1,184	137.5	42.4
	256	−302	2,336	272.7	61.8
RLizard	147	−188	6,176	217.8	165.3
	195	−246	8,240	416.9	232.7
	318	−306	16,448	737.3	382.7
NewHope	112	−213	2,048	329.6	103.6
	257	−216	4,032	673.5	209.2
KYBER	111	−178	1,536	278.2	97.5
	181	−164	2,272	463.6	174.3
	254	−174	3,136	656.0	263.1
SABER	125	−120	1,408	316.9	106.1
	203	−136	2,080	587.6	213.6
	283	−165	2,784	934.8	359.2
LAC	147	−116	1,256	341.2	90.0
	286	−143	2,244	840.1	235.6
	320	−122	2,480	1,101.6	266.6
Round5 (IND-CPA)	128	−88	994	384.4	114.6
	193	−117	1,639	857.2	311.3
	256	−64	2,035	1,794.9	643.4
Threebears	154	−156	1,721	167.8	52.1
	235	−206	2,501	271.4	91.9
	314	−256	3,281	402.5	148.2

• **안전성** 안전성은 4.2절의 방법에 따라 클래식 컴퓨터 환경과 양자컴퓨터 환경에서 알고리즘을 공격하는데 필요한 시간복잡도에 log를 취한 값이다. LizarMong의 Comfort 버전은 NIST에서 제시한 Category1 (AES128bit)의 보안수준을 만족하며 Strong 버전은 Category5 (AES256bit)를 만족한다. Category1 보안수준에서 Newhope, KYBER, SABER는 다소 부족한 것으로 보여진다. LizarMong이 다른 알고리즘들과 달리 192bit 안전성을 보장하는 파라미터를 지원하지 않는다. 하지만, 더 강력한 Strong버전이 다른 알고리즘의 192bit 버전보다 대역폭과 연산속도가 대등한 수준이므로, 실용적 측면에서 문제되지 않는다.

• **정확성** 정확성을 평가하는 방법은 복호화 실패 확률이 얼마나 낮은지 분석하는 것이다. LizarMong은 Comfort버전에서 2^{-179}, Strong에서 2^{-302}의 매우 낮은 실패 확률을 보인다. 특히, 이 연구에서의 계산값은 bit 간의 종속성을 고려한 최신 연구를 반영한 결과임에 주목해야 한다. 나머지 비교 대상 알고리즘들은 모두 bit 간 독립 가정을 두고 계산 한것인데, [8]에 따르면 bit 간 의존성을 고려하면 오류 정정 부호화 기법을 사용하는 LAC의 경우 실패 확률이 2^{48}이나 상승하는것으로 분석될 정도로 큰 영향을 준다. KYBER, SABER, LAC, Round5는 보안수준에 비해 실패확률이 더 낮은 것으로 보인다. 이는 [36]에서 보인 실패 확률을 악용한 공격에 노출될 수 있다. LizarMong은 bit 간 종속성을 가정하에 두고 실패 확률을 계산했음에도 충분한 여유를 두어 실패 확률을 악용한 공격으로부터 자유롭다.

• **대역폭** Key Encapsulation Mechanism은 Alice가 Bob에게 pk를 보내고, Bob은 Alice에게 c를 보내어 각자 보유한 비밀값으로 공유 키를 만들어내는 기법이다. 따라서 pk와 c는 통신 대역폭에 중요한 영향을 주며, c와 pk는 저장하여 상호 간 정확한 값임을 확인해야 하기 때문에 메모리에도 중요한 영향을 준다. 따라서, 대역폭은 공개키와 암호문의 크기로 결정한다. 대역폭은 RLWE 계열 알고리즘에서 매우 중요한 평가 척도이다. 현용 공개 키 암호가 적용되는 네트워크와 프로토콜에 사용할 수 있는 양자내성암호가 필요한데, 현용 공개 키 암호(RSA, ECC)는 대역폭이 1KB 미만인 반면, RLWE계열은 1 $\sim$ 2KB 수준이기 때문이다. LizarMong의 대역폭도

현용 공개 키 암호만큼 작게 설정되지는 못했지만, IND-CCA를 지원하는 KEM 중 가장 적다.(차 순위인 LAC 대비 Comfort과 Strong 각각 약 5% 이상 감소) 참고로, LizarMong의 보안수준은 NewHope 등이 주장한 Category1 보안수준보다 월등히 높다. RLizard의 경량 버전[21]처럼 LizarMong도 보안수준을 NewHope 수준으로 맞출 경우 현용 공개 키 암호 수준의 대역폭을 달성할 것으로 예상할 수 있다.

- **연산속도** LizarMong 연산 속도는 다른 알고리즘에 비해 탁월하게 빠르다 (차 순위인 ThreeBears 대비 Comfort는 1.25배, Strong은 1.65배 향상). 이 결과는 LizarMong이 바탕을 둔 RLizard의 암복호화 속도가 경쟁력이 있었고, 파라미터를 줄이면서 연산량이 적어졌으며, 모듈러스 q를 256으로 설정함에 따라 연산 효율이 매우 높아졌기 때문이다. 특히, 주목해야 할 것은 LizarMong의 연산속도는 3.6절에 기술한 부채널공격 대응기법을 모두 반영한 수치라는 점이다. 다른 알고리즘들이 LizarMong과 같은 대응기법을 반영한다면 오버헤드가 부과되는 것은 명백하기 때문에 LizarMong의 연산속도는 더욱 부각될 수 있다.

제 5 장 결 론

NIST 공모 2라운드 후보 알고리즘을 저마다의 장단점을 갖고 있어 안전성, 정확성, 성능, 대역폭 측면에서 모두 우수한 탁월한 알고리즘을 선택하기 어려운 실정이다. 또한, 최근 활발히 연구되고 있는 부채널 공격에 대한 대응기법 고려가 부족하고 bit 간 종속성을 고려한 복호화 실패확률 계산 및 이를 악용한 공격 등 우회공격에 대한 고려가 필요해졌다. 따라서, 본 연구에서는 모든 측면에서 우수하고 최신 연구들을 반영한 탁월한 알고리즘을 설계하고자 하였다. 본 연구에서 제안한 LizarMong은 Ring-LWE+Ring-LWR 기반의 IND-CCA2 Key Encapsulation Mechanism 및 공개키 암호이다. RLizard를 기본으로하여 256의 작은 모듈러스를 선택하여 XE5 에러 정정 부호화 기법, 공개 키 및 암호문의 부분적 압축기법을 적용하여 대역폭과 연산 속도를 향상시켰다. 또한, 최신 연구를 반영하여 bit 간 종속성을 고려한 복호화 실패 확률을 계산하였다. 다양한 부채널공격으로부터 저항성을 갖도록 CDT 오류 샘플러를 중심 이항 분포 샘플러로 변경하고, 곱셈 연산을 희박한 다항식 곱셈에 하이딩 기법을 적용한 알고리즘 8으로 사용하였다. 결과적으로 LizarMong은 알려진 부채널 공격으로부터 저항성을 갖으며 NIST의 후보 알고리즘 대비 대역폭은 약 5-42% 작으며, 성능은 약 1.2-4.1배 빠르다. 본 연구에서 제시한 LizarMong은 향후 지속적인 안전성 분석과 최신 기술을 접목하여 보다 빠르고 안전하며 적은 대역폭으로도 운영될수 있도록 개선되어야 한다. 이번 연구가 국방 환경에 적합한 양자내성 암호 연구의 기폭제가 되길 바라며 향후 지속적인 연구를 기대한다.

참고 문헌

[1] J. H. Cheon, D. Kim, J. Lee, and Y. Song, "Lizard public key encryption," Technical report, National Institute of Standards and Technology, 2017 . . . , Tech. Rep., 2018.

[2] H. Baan, S. Bhattacharya, S. R. Fluhrer, O. Garcia-Morchon, T. Laarhoven, R. Rietman, M.-J. O. Saarinen, L. Tolhuizen, and Z. Zhang, "Round5: Compact and fast post-quantum public-key encryption." *IACR Cryptology ePrint Archive*, vol. 2019, p. 90, 2019.

[3] M. Mosca, "Cybersecurity in an era with quantum computers: will we be ready?" *IEEE Security & Privacy*, vol. 16, no. 5, pp. 38–41, 2018.

[4] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—a new hope," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 327–343.

[5] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber: a cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2018, pp. 353–367.

[6] J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren, "Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure kem," in *International Conference on Cryptology in Africa*. Springer, 2018, pp. 282–305.

[7] X. Lu, Y. Liu, Z. Zhang, D. Jia, H. Xue, J. He, B. Li, K. Wang, Z. Liu, and H. Yang, "Lac: Practical ring-lwe based public-key encryption with byte-level modulus." *IACR Cryptology ePrint Archive*, vol. 2018, p. 1009, 2018.

[8] J.-P. D'Anvers, F. Vercauteren, and I. Verbauwhede, "The impact of error dependencies on ring/mod-lwe/lwr based schemes," in *International Conference on Post-Quantum Cryptography*. Springer, 2019, pp. 103–115.

[9] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2010, pp. 1–23.

[10] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2012, pp. 719–737.

[11] A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen, "On the hardness of learning with rounding over small modulus," in *Theory of Cryptography Conference*. Springer, 2016, pp. 209–224.

[12] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, p. 34, 2009.

[13] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld, "Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance," *Journal of Cryptology*, vol. 31, no. 2, pp. 610–640, 2018.

[14] T. Fritzmann, T. Pöppelmann, and J. Sepulveda, "Analysis of error-correcting codes for lattice-based key exchange," in *International Conference on Selected Areas in Cryptography*. Springer, 2018, pp. 369–390.

[15] J. W. Bos, S. Friedberger, M. Martinoli, E. Oswald, and M. Stam, "Assessing the feasibility of single trace power analysis of frodo," in *International Conference on Selected Areas in Cryptography*. Springer, 2018, pp. 216–234.

[16] J.-P. D'Anvers, Q. Guo, T. Johansson, A. Nilsson, F. Vercauteren, and I. Verbauwhede, "Decryption failure attacks on ind-cca secure lattice-based schemes," in *IACR International Workshop on Public Key Cryptography*. Springer, 2019, pp. 565–598.

[17] E. Alkim, J. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, D. Stebila *et al.*, "Frodokem: Learning with errors key encapsulation (2019)," *URL: https://csrc. nist. gov/projects/post-quantum-cryptography/round-2-submissions. Citations in this document*, vol. 1, no. 1.3, pp. 1–3.

[18] M. Hamburg, "Three bears," Technical Report. National Institute of Standards and Technology, Tech. Rep., 2017.

[19] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the fujisaki-okamoto transformation," in *Theory of Cryptography Conference*. Springer, 2017, pp. 341–371.

[20] T. Banerjee and M. A. Hasan, "Energy consumption of candidate algorithms for nist pqc standards," Technical report, Centre for Applied Cryptographic Research (CACR) at the . . . , Tech. Rep., 2018.

[21] J. Lee, D. Kim, H. Lee, Y. Lee, and J. H. Cheon, "Rlizard: Post-quantum key encapsulation mechanism for iot devices," *IEEE Access*, vol. 7, pp. 2080–2091, 2018.

[22] C. Peikert, O. Regev, and N. Stephens-Davidowitz, "Pseudorandomness of ring-lwe for any ring and modulus," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2017, pp. 461–473.

[23] J. H. Cheon, D. Kim, J. Lee, and Y. Song, "Lizard: Cut off the tail! practical post-quantum public-key encryption from lwe and lwr," Cryptology ePrint Archive, Report 2016/1126, 2016, https://eprint.iacr.org/2016/1126.

[24] A. Park and D.-G. Han, "Chosen ciphertext simple power analysis on software 8-bit implementation of ring-lwe encryption," in *2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST)*. IEEE, 2016, pp. 1–6.

[25] B. Applebaum, D. Cash, C. Peikert, and A. Sahai, "Fast cryptographic primitives and circular-secure encryption based on hard learning problems," in *Annual International Cryptology Conference*. Springer, 2009, pp. 595–618.

[26] P. Ravi, D. B. Roy, S. Bhasin, A. Chattopadhyay, and D. Mukhopadhyay, "Number "not used" once-practical fault attack on pqm4 implementations of nist candidates," in *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2019, pp. 232–250.

[27] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, gauss, and reload–a cache attack on the bliss lattice-based signature scheme," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 323–345.

[28] S. Akleylek, E. Alkım, and Z. Y. Tok, "Sparse polynomial multiplication for lattice-based cryptography with small complexity," *The Journal of Supercomputing*, vol. 72, no. 2, pp. 438–450, 2016.

[29] M. Walters and S. S. Roy, "Constant-time bch error-correcting code." *IACR Cryptology ePrint Archive*, vol. 2019, p. 155, 2019.

[30] M.-J. O. Saarinen, "Hila5: On reliability, reconciliation, and error correction for ring-lwe encryption," Cryptology ePrint Archive, Report 2017/424, 2017, https://eprint.iacr.org/2017/424.

[31] H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma, "Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited," in *Annual International Cryptology Conference*. Springer, 2018, pp. 96–125.

[32] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[33] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.

[34] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer, "Revisiting the expected cost of solving usvp and applications to lwe," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 297–322.

[35] M. Albrecht, "A sage module for estimating the concrete security of learning with errors instances (2017)."

[36] J.-P. D'Anvers, F. Vercauteren, and I. Verbauwhede, "On the impact of decryption failures on the security of lwe/lwr based schemes," Cryptology ePrint Archive, Report 2018/1089, 2018, https://eprint.iacr.org/2018/1089.

[37] S. Fluhrer, "Cryptanalysis of ring-lwe based key exchange with key share reuse," Cryptology ePrint Archive, Report 2016/085, 2016, https://eprint.iacr.org/2016/085.

[38] S. Kim and S. Hong, "Single trace analysis on constant time cdt sampler and its countermeasure," *Applied Sciences*, vol. 8, no. 10, p. 1809, 2018.

[39] R. Primas, P. Pessl, and S. Mangard, "Single-trace side-channel attacks on masked lattice-based encryption," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2017, pp. 513–533.

[40] A. Aysu, Y. Tobah, M. Tiwari, A. Gerstlauer, and M. Orshansky, "Horizontal side-channel vulnerabilities of post-quantum key exchange protocols," in *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. IEEE, 2018, pp. 81–88.

[41] W.-L. Huang, J.-P. Chen, and B.-Y. Yang, "Correlation power analysis on ntru prime and related countermeasures." *IACR Cryptology ePrint Archive*, vol. 2019, p. 100, 2019.

[42] J. W. Bos, S. Friedberger, M. Martinoli, E. Oswald, and M. Stam, "Assessing the feasibility of single trace power analysis of frodo," in *International Conference on Selected Areas in Cryptography*. Springer, 2018, pp. 216–234.

[43] T. Espitau, P.-A. Fouque, B. Gerard, and M. Tibouchi, "Loop-abort faults on lattice-based signature schemes and key exchange protocols," *IEEE Transactions on Computers*, vol. 67, no. 11, pp. 1535–1549, 2018.

[44] O. Reparaz, S. S. Roy, F. Vercauteren, and I. Verbauwhede, "A masked ring-lwe implementation," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2015, pp. 683–702.

[45] O. Reparaz, R. de Clercq, S. S. Roy, F. Vercauteren, and I. Verbauwhede, "Additively homomorphic ring-lwe masking," in *Post-Quantum Cryptography*. Springer, 2016, pp. 233–244.

[46] T. Oder, T. Schneider, T. Pöppelmann, and T. Güneysu, "Practical cca2-secure and masked ring-lwe implementation," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 142–174, 2018.

[47] J. Howe, A. Khalid, M. Martinoli, F. Regazzoni, and E. Oswald, "Fault attack countermeasures for error samplers in lattice-based cryptography," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

Abstract

Design and Implementation of Post-Quantum Cryptography for Military

Chi-Gon Jung
Department of Engineering Practice
Graduate School of Engineering Practice
Seoul National University

As the threat of public-key cryptography by a quantum-computer is emerging, NIST is going on a post-quantum cryptography standardization process. On the other hand, In the military, it is an urgency to adopt post-quantum cryptography, but related studies are insufficient. Civilians can switch to post-quantum cryptography without deploy issues using international standard algorithms selected through the NIST competition. But the military can not easily. Because the military requires algorithms to be applied to various weapons and communication systems, excellent algorithms are required in terms of performance, bandwidth, correctness, and security. Besides, there is a lack of countermeasures against the side-channel attack that considers lost cryptographic equipment during military operations. Also, legislation that allows only domestic cryptography to be used to protect military secrets imposes restrictions on the use of international standard algorithms selected by NIST. Therefore, we propose a Key Encapsulation Mechanism and public-key encryption scheme suitable military. It called LizarMong, which is based on RLizard. LizarMong combines the merit of NIST's candidate algorithms and state-of-the-art studies such as countermeasures against known side-channel attacks. As a result, it achieves up to 85% smaller band-

width and 3.3 times faster performance compared with RLizard. Compared with the NIST's candidate algorithms with a similar security, the bandwidth is about 5-42% smaller, and the performance is about 1.2-4.1 times faster. Also, LizarMong resists the known side-channel attacks.

Keywords: Ring Learning with error, post-quantum cryptography

Student Number: 2018-26518