효과적인 하드웨어 설계를 위한 격자기반 양자내성암호 통합 모듈 가속 기법 연구

이용석 [1], 백윤흥 [1]
[1] 서울대학교 전기정보공학부, 서울대학교 반도체 공동연구소
yslee@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on Integrated Module Acceleration of Lattice-based Post Quantum Cryptography for Efficient Hardware Design

Yongseok Lee[1], Yunheung Paek[1]
[1]Dept. of Electrical and Computer Engineering and Inter-University Semiconductor Research Center(ISRC), Seoul National University

요 약

양자내성암호(PQC, Post Quantum Cryptography)는 기존 컴퓨터 시스템과 양자 컴퓨터 시스템에서 모두 보안성을 보장하는 암호체계이다. NIST 에서 표준화로 선정된 네 가지 알고리즘을 보면, 그 중 세 가지의 알고리즘이 모두 격자기반 알고리즘임을 알 수 있다. 이에 따라 격자기반 알고리즘에서 공통되는 연산을 통합하여 효과적으로 구현하려는 연구들이 진행되고 있으며, 그 중에는 RISC-V 를 통한 하드웨어/소프트웨어 통합 설계 연구 그리고 FPGA 혹은 ASIC 을 이용한 하드웨어 통합 모듈 가속 연구들이 있다. 본 논문에서는 효과적인 하드웨어 설계를 위해 세 종류의 격자기반 양자내성암호 연산을 분석하고, 이를 하드웨어에서 효과적으로 구현한 기존 연구 사례를 분석한다. 이를 통해 통합 모듈 가속 기법 연구를 위해 고려되는 면적 대비 연산 성능 그리고 면적 대비 소비전력 등의 수치를 높이는 과정에서 발생하는 문제점과 이를 효과적으로 극복하는 방법들을 소개한다.

1. 서론

기존 컴퓨터 시스템 뿐만 아니라 양자 컴퓨터 시스템에서도 보안성을 보장하기 위해 개발된 양자내성암호(PQC, Post Quantum Cryptography) 알고리즘은 해시기반, 다변수기반, 코드기반, 격자기반 등 다각도로 개발되고 있다[1]. 특히 미국 NIST(National Institute of Standard and Technology)에서는 2017 년부터 현재까지 양자내성암호 표준화를 진행하고 있으며, 이러한 네 가지 기반을 바탕으로 알고리즘들을 분류하고 있다[2]. 양자내성암호 알고리즘의 타입은 크게 두 가지 분야로 나눌 수 있으며, 이에따라 KEA(Key Encapsulation Algorithm) 그리고 DSA(Digital Signature Algorithm)를 구분하여 표준화를 진행하고 있다. 최근 표준화 알고리즘 선정된 네 가지 알고리즘은 KEA 분야에서 Crystal-Kyber[3], 그리고 DSA 분야에서 Crystal-Dilithium[4], FALCON[5], SPHINCS+[6]를 선정하였다. 이렇게 선정된 알고리즘들은 표준화 문서 초안(FIPS203, FIPS204, FIPS205)을 발표하며 표준화 작업을 진행하며 마무리 수준에 있다고 볼 수 있다. 또한 알고리즘의 다양성을 위해, 이와 별도로 KEA/DSA 분야에 대해 KEA 라운드 4 그리고 DSA 추가 라운드 1 으로 알려진 추가적인 표준화 라운드도 진행중에 있다.

표준화 과정에서 선정된 네 가지의 알고리즘을 살펴보면, 네 가지 중에서 세 가지에 해당하는 Crystal-Kyber, Crystal-Dilithium, FALCON 알고리즘들이 모두 격자기반 알고리즘이며, 오직 SPHINCS+ 알고리즘만 해시기반 알고리즘임을 알 수 있다. 표준화 알고리즘으로 하나의 알고리즘이 선정되는 것이 아니라, 여러 개의 알고리즘이 동시에 선정되었고, 이는 양자내성암호를 사용하는 기업이나 사용자 입장에서 선택적으로 알고리즘을 사용할 수 있는 환경이 필요하다고 할 수 있다. 현재 사용되는 암호체계인 RSA, ECC 알고리즘 등도 사용되는 환경에 맞게 여러 알고리즘의 기능을 동시에 구현하기도 하고, 선택적으로 사용하는 방식이 사용되고 있다.

이런 환경에 맞춰 최근 격자기반 알고리즘들에 대해 공통된 연산을 통합하여 하나의 블록에서 효과적으로 구현하려는 연구들이 진행되고 있다. 하지만 격

자기반 알고리즘들도 모두 같은 격자를 가정하고 있는 것이 아니며, 사용되는 모듈러 연산의 프라임(Prime) 비트도 서로 다른 특징이 있다.

따라서 이런 연산에 대한 유연성을 가질 수 있는 것을 목표로 하는 통합 연구가 진행되고 있으며, [7] 연구는 RISC-V 라는 임베디드 환경에서 하드웨어/소프트웨어 통합 설계로 Kyber 와 Dilithium 알고리즘을 모두 수행할 수 있는 방법을 제안하였다. 이는 서로 다른 격자기반 알고리즘의 모듈러 연산을 효과적으로 통합하는 방법으로 기존 소프트웨어 단독 수행 대비 약 10 배 빠른 연산 속도를 보여주었다. 이와 다른 방식으로 [8] 연구는 하드웨어 통합 모듈 설계를 통해 Kyber 와 Dilithium 알고리즘을 모두 수행할 수 있는 모듈러 연산 모듈을 제안하였다. 이는 통합 모듈이 기존의 단독 설계 모듈보다도 면적대비 효과적인 성능을 보일 수 있음을 보여주었다.

이처럼 격자기반 알고리즘들은 서로 다르지만 공통되면서 비슷한 연산을 수행하는 모듈이 존재하며, 이를 통합해서 효과적으로 가속하려는 연구들이 진행되었다. 본 논문에서는 효과적인 하드웨어 설계를 위한 통합 모듈 가속 기법들을 중심으로 분석하려 한다. 먼저 세 가지의 격자기반 양자내성암호 알고리즘들이 가지는 연산 특징들을 분석하고, 이를 효과적으로 통합하는 과정에서 발생하는 문제점 그리고 기존 연구에서 극복한 기법들을 면적 대비 성능 혹은 면적 대비 소비전력 측면에서 분석한다. 이를 통해 격자기반 양자내성암호 알고리즘들에 대해 효과적인 통합 모듈 기법 설계하는 방법을 제안한다.

2. 격자기반 양자내성암호 알고리즘 특징

양자내성암호 중 격자기반 알고리즘은 그림 1 과 같이 격자구조에서 발생하는 SVP(Shortest Vector Problem) 혹은 CVP(Closest Vector Problem)에 기초하고 있다. 하지만 표 1 에서 확인할 수 있듯이, 격자구조 타입도 Crystal-Kyber 와 Crystal-Dilithium 알고리즘에서 사용하는 모듈격자 그리고 FALCON 알고리즘에서 사용하는 Ideal 격자가 다른 특징이 있다. 이는 다항식 변환 방식에도 차이를 가져오는데, Crystal-Kyber 와 Crystal-Dilithium 알고리즘은 NTT(Number Theoretic Transform)와 INTT(Inverse Number Theoretic Transform)를 사용해서 다항식 변환을 수해하는 것과 달리, FALCON 알고리즘에서는 NTT/INTT 는 물론이고 FFT(Fast Fourier Transform)와 IFFT(Inverse Fast Fourier Transform)도 사용한다. 특히 NTT/INTT 가 정수형 연산을 수행하는 것과 달리, FFT/IFFT 는 실수형 연산을 수행하여 일반적으로 하드웨어 환경에서 정수형 연산보다 더 많은 연산 시간을 초래하는 특징이 있다.

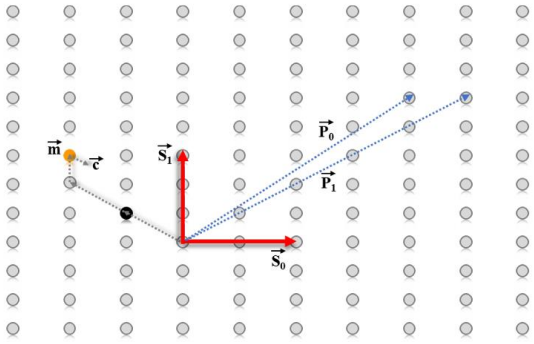


그림 1. 격자기반 암호에서 2 차원 격자구조 예시[2].

표 1. 격자기반 양자내성암호 알고리즘에 따른 격자 타입, 사용되는 변환 연산, 데이터 비트 수.

Algorithm Type	Algorithm	Lattice Type	Transform	Data Bits
KEA	CRYSTAL-Kyber	Module Lattices	NTT/INTT	12-bit
DSA	CRYSTAL-Dilithium	Module Lattices	NTT/INTT	23-bit
	FALCON	Ideal Lattices	NTT/INTT FFT/IFFT	64-bit

데이터 비트 수를 비교해보면, 이 또한 세 알고리즘이 모두 다른 데이터 비트 수를 사용하고 있는 것을 알 수 있다. 이는 소프트웨어 환경에서 수행되는 가속 연구에서는 크게 영향이 없을 수 있으나, 하드웨어 설계를 통한 모듈 연산에서는 데이터 비트 수에 따라서 설계되는 모듈의 타겟 비트 수와 데이터 통신을 위한 비트너비가 달라진다.

예를 들어 23 비트너비로 메모리와 연산 모듈 간의 데이터 인터페이스를 연결할 경우, 23 비트 연산을 사용하는 Crystal-Dilithium 알고리즘에서는 모든 하드웨어 자원의 사용량(Utilization)이 높게 나타나지만, 같은 하드웨어 환경에서 12 비트 연산을 사용하는 Crystal-Kyber 알고리즘을 수행할 경우 (23-12)비트에 해당하는 나머지 11 비트 너비의 하드웨어 자원을 사용되지 않고 낭비되게 된다. 또한 64 비트 연산을 사용하는 FALCON 알고리즘의 경우 이러한 하드웨어 환경에서 연산이 불가능하거나, 23 비트 단위로 3 번에 걸쳐 분할하여 연산하여 더 오랜 연산 시간을 소요할 수 있다는 하드웨어 비효율적 설계 문제점이 발생할 수 있다.

3. 하드웨어 통합 모듈 가속 연구

격자기반 양자내성암호 알고리즘들의 하드웨어 가속 연구들은 2 장에서 살펴본 것처럼 각 알고리즘별

연산특징들로 인해 개별적인 최적화 연구[9]가 주로 진행되어 왔다. 하지만 통합 설계를 통한 연구들 [7,8,10,11]도 새롭게 진행되고 있으며, 각각 연구들의 특징을 분석하려 한다.

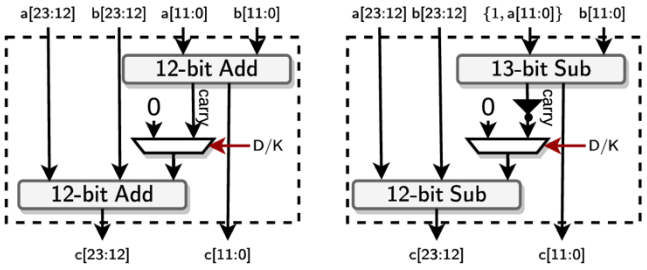
표 2. Crystal-Kyber 와 Crystal-Dilithium 에서 알고리즘별 단독 설계 하드웨어와 알고리즘 통합 설계 하드웨어의 성능 비교.

Ref.	Algo.	Area (KGE)	Tech. (nm)	Latency (us)	Freq. (MHz)
[9]	K	104	65	160	200
[10]	D/S	854	65	182/-	400
[7]	K/D	149	28	2,209 /5,446	200
[8]	K/D	769	65	90/262	280/560

*항목 약어 표현: (Ref.: Reference, Algo.: Algorithms, Tech.: Technology, Freq.: Frequency).
*알고리즘 약어 표현: (K: Crystal-Kyber, D: Crystal-Dilithium, S: Sabor).

먼저, [9] 연구에서는 Crystal-Kyber 알고리즘에 대한 하드웨어 가속 연구를 ASIC 환경에서 수행했다. 이는 Security Level 5 를 충족하는 Kyber-1024 파라미터에서 Encapsulation 함수와 Decapsulation 함수를 모두 수행하는데 160us 의 성능을 보이면서도, 65nm 공정으로 합성하였을 때 104KGE 의 설계면적을 차지하였다. 또한 [10] 연구에서는 Crystal-Dilithium 알고리즘에 대한 하드웨어 가속 연구로 65nm 공정의 ASIC 환경으로 설계했을 때 Security Level 3 를 충족하는 Dilithium-3 파라미터에서 서명 생성 및 검증 시간이 총 182us 가 걸렸으며, 설계면적은 854KGE 를 차지하였다.

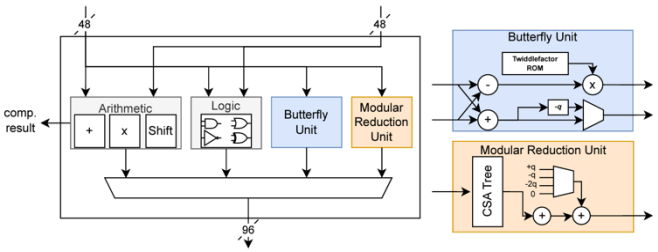
그림 2. 23 비트 연산과 12 비트 연산이 통합된 모듈 예시[8]. D: Crystal-Dilithium, K: Crystal-Kyber.



최근의 두 연구[7,8]에서는 Crystal-Kyber 와 Crystal-Dilithium 알고리즘을 통합하는 연구를 수행하였는데, [7] 연구에서는 RISC-V 를 통한 하드웨어/소프트웨어 통합 설계 방법을 적용하였고, [8] 연구는 하드웨어 단독 설계를 수행하였다. 이에 [7] 연구에서는 RISC-V 에 통합한 특성으로 설계면적은 149KGE 로 적게 차지하였지만, 연산 성능이 다른 연구들에 비해 높게 나타났다. 이와 다르게 하드웨어 단독으로 통합한 [8] 연구에서는 효과적으로 Crystal-Kyber 와 Crystal-Dilithium 알고리즘을 하나의 통합 모듈에서 가속하기 위한 설계 방법을 제시하였다. 이러한 방식은 개별적인 설계 모듈을 배치하여 여러 알고리즘을 하드웨어로 구현하는 것과 달리 공통된 연산을 수행할 수 있는 통합 모듈을 만드는 것이다. 그림 2 에서 볼 수 있듯이, 24 비트 데이터 인터페이스를 사용해서 23 비트 데이터는 하나씩 불러오고, 12 비트 데이터는 두 개씩 패킹하여 불러오는 방식을 사용한다. 또한 연산 모듈을 12 비트 단위로 배치하여 중간에 캐리 비트를 조절하는 방식으로 23 비트 연산 하나를 수행할 지 12 비트 연산 두 개를 수행할 지 선택할 수 있다. 이 같은 방법을 사용하면, 데이터 인터페이스에서 낭비되는 자원을 최소화할 수 있으며, 실제 연산 모듈에 대해서도 낭비되는 자원을 최소화 하면서 연산이 가능해진다. 하지만 이러한 통합 모듈 가속 방식은 두 알고리즘 사이의 성능 차이를 발생시키는 문제점이 있다.

같은 연산 모듈을 사용한다면, 23 비트 데이터 연산을 사용하는 Cyrstal-Dilithium 연산이 한 번 수행될 때 12 비트 데이터 연산을 사용하는 Crystal-Kyber 연산은 두 번 수행되며, 같은 연산량을 가지는 파라미터에서는 Crystal-Kyber 알고리즘이 두 배 더 빠른 연산 성능을 보이게 된다.

그림 3. 연산 성능 밸런스를 위한 통합 모듈 가속 설계 기법 예시[11].



이처럼 통합된 알고리즘 사이의 성능 밸런스가 맞지 않는 문제점을 해결하기 위해, [11] 연구에서는 알고리즘별 연산 함수들을 프로파일링하여 하드웨어 연산 모듈의 비율을 다르게 배치하는 방식을 제안하였다. 이는 그림 3 에서 볼 수 있듯이 통합된 알고리즘들이 모두 비슷한 연산 성능을 발휘하는 것을 목표로 연산 모듈의 비율을 조정하여 구성하는 방법이다. 이는 15nm 공정의 ASIC 에서 611KGE 설계면적을 차지하면서 [8] 연구와 평균적으로는 비슷한 연산 성능을 보여주었지만, 두 알고리즘 사이의 성능 밸런스를 맞췄다는 점에서 하드웨어 통합 모듈 설계에 적합한 효과적인 가속 연구이다.

4. 결론

본 논문에서는 격자기반의 양자내성암호들의 특징을 알아보고, 이를 하드웨어로 통합 모듈 가속하는 연구들에 대해 각각 분석하였다. 이는 기존 알고리즘 개별 최적화에서 고려하지 않았던 설계 모듈들의 어려움을 해결하는 과정에서 연산 모듈의 활용성과 데이터 인터페이스의 활용성을 효과적으로 높이는 기법들을 적용하였다. 또한 그러면서 발생하는 통합된 알고리즘들 사이의 성능 밸런스가 새로운 이슈로 작용하였고, 이를 해결하기 위해 통합 모듈들의 비율을 조정하여 최종적인 연산 성능 밸런스를 고려한 통합 모듈 설계 방법에 대해서도 분석하였다. 이러한 설계 방법은 다양한 격자기반 알고리즘을 통합하여 사용할 경우 고려해야 하는 문제점들을 잘 보여주고 있으며, 향후 양자내성암호 알고리즘의 하드웨어 연구에 좋은 기여가 될 것으로 기대된다.

ACKNOWLEDGEMENT

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구이며 (IITP-2023-RS-2023-00256081), 2024 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이며 (RS-2023-00277326), 2024 년도 정부(산업통상자원부)의 재원으로 한국산업기술기획평가원의 지원을 받아 수행된 연구이며(No. RS-2024-00406121, 자동차보안취약점기반 위협분석시스템개발(R&D)), 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구이며(No.RS-2024-00438729, 익명화된 기밀실행을 이용한 전주기적 데이터 프라이버시 보호 기술 개발), 2024 년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음.

참고문헌

[1] Nejatollahi, Hamid, et al. "Post-quantum lattice-based cryptography implementations: A survey." *ACM Computing Surveys (CSUR)* 51.6 (2019): 1-41.

[2] Asif, Rameez. "Post-quantum cryptosystems for Internet-of-Things: A survey on lattice-based algorithms." *IoT* 2.1 (2021): 71-91.

[3] Avanzi, R. et al. CRYSTALSKyber: Algorithm Specifications and Supporting Documentation, Submission to the NIST Post-Quantum Project. 2021. Available online: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf

[4] Ducas, L. et al. CRYSTALS-Dilithium: Algorithm Specifications and Supporting Documentation, Submission to the NIST Post-Quantum Project. 2021. Available online: https: //pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf

[5] Fouque, P.A. et al. Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU, Specification v1.2. 2020. Available online: https: //falcon-sign.info/falcon.pdf

[6] Aumasson, J.P. et al. SPHINCS+ Specification. Submission to the NIST Post-Quantum Project. 2020. Available online: https://sphincs.org/data/sphincs+-r3.1-specification.pdf

[7] Ye, Zewen, et al. "A Highly-efficient Lattice-based Post-Quantum Cryptography Processor for IoT Applications." *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2024.2 (2024): 130-153.

[8] Aikata, Aikata, et al. "KaLi: A crystal for post-quantum security using Kyber and Dilithium." *IEEE Transactions on Circuits and Systems I: Regular Papers* 70.2 (2022): 747-758.

[9] Bisheh-Niasar, et al. "Instruction-set accelerated implementation of CRYSTALS-Kyber." *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.11 (2021): 4648-4659.

[10] Aikata, Aikata, et al. "A unified cryptoprocessor for lattice-based signature and key-exchange." *IEEE Transactions on Computers* 72.6 (2022): 1568-1580.

[11] Jung, Heonhui, et al. "Designing a Scalable and Area-Efficient Hardware Accelerator Supporting Multiple PQC Schemes." *Electronics* 13.17 (2024): 3360.