NIST 양자내성암호 공모전 KEM Finalist 최적 구현 동향

권혁동* 엄시우** 심민주** 서화정***

*한성대학교 정보컴퓨터공학과 (대학원생)
**한성대학교 IT융합공학부 (대학원생)
***한성대학교 IT융합공학부 (조교수)

Optimized Implementation Trend of NIST Post Quantum Cryptography Competition KEM Finalist

Hyoek-Dong Kwon* Si-Woo Eum** Min-Joo Sim** Hwa-Jeong Seo***

*Hansung University Dept. of Information Computer Engineering (Graduate student)
**Hansung University Dept. of IT convergence Engineering (Graduate student)
***Hansung University Dept. of IT convergence Engineering (Assistant Professor)

요 약

양자컴퓨터의 발전에 따라 기존 대칭키, 공개키 알고리즘 체계가 위협을 받고 있다. 이에 따라 양자컴퓨터 환경 상에서도 안전하게 사용 가능한 양자내성암호의 개발이 활발하게 이루어지고 있다. 특히 2017년 미국 국립표준기술연구소는 양자컴퓨터 환경에 대비하기 위해 양자내성암호 표준화 공모전을 개최하였다. 양자내성암호는 양자컴퓨터 환경에서 안전하게 사용될 것으로 전망되나, 효과적으로 가동하기 위해서는 추가적인 연구가 필요한 실정이다. 본 논문에서는 양자내성암호 공모전의 Finalist 진출 알고리즘 중 KEM 부문의 최적 구현 동향에 대해서 알아본다.

I. 서론

양자컴퓨터는 양자역학의 원리를 사용하여 고전컴퓨터에 비해 빠른 연산 속도를 가지는 새로운 유형의 컴퓨터로 Feynman의 제안에서 비롯되었다[1]. 양자컴퓨터 상에서는 검색을 빠르게 하는 Grover 알고리즘[2], 소인수 분해에 최적화된 Shor 알고리즘 등[3]이 구현이 가능하며 각각 brute-force와 소인수 분해를 빠르게 진행할 수 있다. 따라서 현재 사용 중인 대칭키, 공개키 기반의 암호 알고리즘들을 양자컴퓨터 환경에서는 사용하기 위험할 것으로 예상된다.

이러한 상황을 극복하기 위해서 양자내성암호의 개발이 활발하게 이루어지고 있다. 본 논문에서는 NIST에서 주관 및 진행 중인 양자내성암호 공모전의 알고리즘 중, Finalist 진출 알고리즘의 최적 구현 동향에 대해서 알아본다.

II. 양자내성암호 공모전

2017년 NIST(미국 국립표준기술연구소, National Institute of Standards and Technology)는 양자내성암호 표준화를 위한 공모전을 개최하여 2020년에 Round 3 Final까지 진행하였다. 공모전은 크게 공개키 암호화 부문과 전자 서명 분야로 나뉘어 진행하고, 대체 후보군을 제외하고 각각 4종, 3종의 알고리즘이 진출하였다[4]. 공개키 부분의 알고리즘은 Classic McEliece, CRYSTALS-KYBER, NTRU, SABER가 있으며, 전자서명 부문에는 CRYSTALS-DILITHIUM, FALCON, Rainbow가 있다[5].

III. 최적 구현 동향

3.1 Classic McEliece

Classic McEliece는 1979년에 제안된 알고리즘으로, 양자내성암호 공모전에는 부호 기반 알고리즘으로 진출하였다[5]. Classic McEliece는 Round 3 Finalist 중 가장 작은 암호문 크기를 가지고 있다. 마지막으로 처음으로 제안된 지 아주 오랜 시간이 흘렀으며 그 동안 다양한 공격과 검증이 이루어졌기에 신뢰성과 안전성이 높다는 특징이 있다.

[6]은 Classic McEliece를 ARM Cortex-M4 프로세서 상에서 구현을 시도하였다. 제안하는 기법은 constant-time 구현을 하였고, 대상 플랫폼인 STM32F4-Discovery에서 SRAM이 부족하여 공개키를 저장하기 어려운 문제를 flash memory에 저장하여 해결하는 기법을 제시하였다. 구현 결과, 레벨 1 매개변수 세트는 FrodoKEM의 레벨 1 매개변수 세트와 비교해서 Encapsulation은 80배 이상, Decapsulation은 17배 이상 빠르다.

[7]은 Classic McEliece의 Key Pair Generation 과정을 최적화하였다. 구현에는 AVX(Advanced Vector eXtensions)를 사용하였으며, polynomial multiplication, Gaussian reduction 그리고 linear map 연산 및 활용 부분을 최적화하였다. 특히 곱셈 부분에서는 일부 연산 순서와 구조를 변경하여 명령어 사용을 줄이는 것으로 연산 속도를 증가시켰다. 기존대비 곱셈기의 성능은 최대 75% 가량 향상되었으며, 전체적인 성능 차이는 표 1과 같다.

Table 1. Performance comparison table, over 2,000 tests. (Unit. ms)

Algorithm	Original	Optimized	Percentage
PK-75%	13,421	4,675	65.16%
SK-75%	127.750	67,702	47.00%
KPG-75%	212.542	118.116	55.57%

3.2 CRYSTALS-KYBER

CRYSTALS-KYBER는 전자서명 부문의 Dilithium과 쌍을 이루는 알고리즘으로, CRYSTALS의 공개키 부분은 KYBER로 제안되었다. Kyber는 Ring-LWE(Learning With Errors)를 사용한 Module-LWE 기반의 알고리즘으로 유일하게 LWE를 사용한 Finalist 알고리즘이다[8].

[9]는 Cortex-M4 상에서 하드웨어 구현을 통해 NTT(Number Theoretic Transform)과 Inverse NTT, polynomial multiplication 과정을 최적화 하였다. 제안하는 기법은 세 가지 알고리즘에 통합적으로 사용 가능한 새로운 유형의 Butterfly structure를 구성하였으며, 이를 Key generation, Encapsulation, Decapsulation에 사용할 수 있도록 하였다. 제안하는 기법은 16개의 Butterfly unit을 사용하여 NTT, Inverse NTT 및 polynomial multiplication 알고리즘의 연산 성능을 기존 대비 최대 각각 112배, 132배 109배 향상시켰다.

[10]은 Kyber 알고리즘을 Cortex-A74 프로세서와 Apple M1 프로세서 상에서 구현하였다. 두 프로세서는 ARMv8에 속하는 프로세서로 제공되는 Instruction Set이 동일하기에 같은 기법을 적용하였다. 제안하는 기법은 Barret multiplication과 Montgomery multiplication을 ARMv8에서 제공하는 병렬 instruction 연산자인 NEON을 사용하여 구현하였다. 표 2는 곱셈 종류와 상황에 따라 구현에 사용된 명령어의 숫자를 나타낸다. 제안하는 곱셈기를 Kyber에 적용한 결과는 표 3과 같이 정리할 수 있다.

Table 2. Number of instructions for optimized implementation of multiplication.

Multiplication	Type	Instructions
Barret	Normal	3
Polynomial	Normal	5
	Constant	4
	Round off	3
	Big number	7

Table 3. Results of optimized Kyber. K: Key pair generation, E: Encapsulation, D: Decapsulation. (Unit: kilo clock cycles)

Algorithm	Cortex A-72			Apple M1		
	K	E	D	K	E	D
Kyber512	62.5	80.7	76.4	14.9	34.8	20.9
Kyber768	99.2	127.5	120.7	23.8	36.3	31.0
Kyber1024	156.7	192.3	184.2	33.0	48.9	44.0

3.3 NTRU

NTRU는 격자 기반 암호의 알고리즘으로 Ring 구조를 활용하는 알고리즘이다[11]. NTRU는 격자 기반 암호 중 제안된 지 오래된 알고리즘에 속하며 그로 인해 다른 알고리즘에 비해 신뢰성이 확보되어있다[12].

[13]에서는 ARMv8의 NEON을 사용하여 NTRU를 Apple M1과 Cortex-A72 프로세서 상에서 구현하였다. 제안하는 기법은 NTT를 레벨 별로 다르게 구현하였고 vector register에 호출되는 값을 최적화하여 효과적인 병렬 연산이 이루어지도록 하였다. 전체적인 성능 비교는 표 4와 같다. 표 4에서 Reference 구현물은 Apple M1 프로세서 상에서 C 코드를 사용하여 가동하였고, AVX2는 AMD EPYC7742 프로세서 상에서 가동한 결과물이다. 구현 결과 Encapsulation은 NTRU-HPS에서 약 3.05~3.24배의 성능 향상을, NTRU-HRSS에서 약 6.68배의 성능 향상이 있었다. Decapsulation에서 NTRU-HPS는 약 7.89~8.49배, NTRU-HRSS는 7.24배의 성능 향상을 보였다.

Table 4. Execution time of NTRU. E; Encapsulation, D: Decapsulation

Algo.	Type	HPS677	HRSS701	HPS821
Ref	E	183.1	152.4	245.3
	D	430.4	439.9	586.5
NEON	E	60.1	22.8	75.7
	D	54.6	60.8	69.0
AVX2	E	26.0	20.4	29.9
	D	75.7	47.7	57.3

3.3 SABER

Saber는 격자 기반 암호 알고리즘으로, Moulde-LWR(Learning With Rounding) 방식을 사용한다[14]. Saber는 Round 3 Finalist 중 유일하게 LWR 방식을 사용하는 알고리즘이다.

[10]은 Kyber에 사용한 최적화 기법을 Saber에도 적용하여 구현하였다. 구현에 차이점이 있다면, Saber는 16-bit용 NTT 구현 및 연산자를 사용하지 않았으며, 32-bit용 NTT 구현 및 연산자를 사용하였다[15]. 제안하는 기법의 Saber의 최적 구현 결과는 표 5와 같다.

Table 5. Results of optimized SABER. K: Key pair generation, E: Encapsulation, D: Decapsulation. (Unit: kilo clock cycles)

Algorithm	Cortex A-72			Apple M1		
	K	E	D	K	E	D
Saber	109.2	140.1	147.9	32.9	44.9	44.1
Light saber	64.2	87.3	92.8	20.1	29.7	28.6
Fire saber	175.1	211.4	222.3	50.3	65.4	64.6

IV. 결론

본 논문에서는 양자내성암호 공모전 Round 3 Finalist 중, KEM 부문 알고리즘의 최적 구현 동향에 대해서 확인하였다. 양자컴퓨터 시대에 대비하여 양자내성암호의 원활한 보급 및 사용을 위해서는 지속적인 연구가 필요할 것으로 예상된다.

V.Acknowledgement

이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임(No.2018-0-00264, IoT 융합형 블록체인 플랫폼 보안 원천 기술 연구, 50%) 그리고 이 성과는 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2020R1F1A1048478, 50%)

[참고문헌]

[1] R.P.Feynman, “Simulating physics with computers,” International Journal of Theoretical Physics, 21, pp.467 - 488 Jun, 1982.

[2] S.Jaques, M.Naehrig, M.Roetteler, and F.Virdia, “Implementing Grover oracles for quantum key search on AES and LowMC,” In Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Cham, pp.280-310, May, 2020.

[3] P.W.Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” In Proceedings 35th annual symposium on foundations of computer science, Ieee, pp.124-134, Nov, 1994.

[4] M.Kumar, and P.Pattnaik, “Post Quantum Cryptography (PQC)-An overview,” *In 2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1-9, Sep, 2020.

[5] Y.S.Kim, “Comparison of characteristics of NIST Post-Quantum Cryptogrpahy Standardization Competition Round 3 algorithms,” Online. [Available]: https://www.itfind.or.kr/comm/binView.do?path=/ResourceData/DataFiles//WZIN/jugidong/1976/&fileName=file2983852861076235638-197602.pdf

[5] R.J.McEliece, “A public-key cryptosystem based on algebraic,” *Coding Thv*, pp. 114-116, 1978.

[6] M.S.Chen, and T.Chou, “Classic McEliece on the ARM Cortex-M4,” IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 125-489, 2021.

[7] M.Ceria, A.De Piccoli, M.Tiziani, and A.Visconti, “Optimizing the Key-Pair Generation Phase of McEliece,” *In 4th International Conference on Wireless, Intelligent and Distributed Environment for Communication*, pp. 111-122, 2022.

[8] R.Avanzi, J.Bos, L.Ducas, E.Kiltz, T.Lepoint, V.Lyubashevsky, and D.Stehlé, “CRYSTALS-Kyber algorithm specifications and supporting documentation,” *NIST PQC Round 3 submission*, Oct, 2020.

[9] F.Yarman, A.C.Mert, E.Öztürk, and E.Savaş, “A hardware accelerator for polynomial multiplication operation of CRYSTALS-KYBER PQC scheme,” *In 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1020-1025, 2021.

[10] H.Becker, V.Hwang, M.J.Kannwischer, B.Y.Yang, and S.Y.Yang, “Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1,” *Cryptology ePrint Archive*, Nov, 2021.

[11] C.Chen, O.Danba, J.Hoffstein, A.Hülsing, J.Rijneveld, J.M.Schanck, P.Schwabe, W.Whyte, and Z.Zhang, “NTRU: Algorithm Specifications And Supporting Documentation,” N*IST PQC Round 3 submission*, Sep, 2020.

[12] J.Hoffstein, J.Pipher, and J.H.Silverman, “NTRU: A ring-based public key cryptosystem,” *In International algorithmic number theory symposium*, pp. 267-288, 1998.

[13] D.T.Nguyen and K.Gaj. “Optimized software implementations of CRYSTALS-Kyber, NTRU, and Saber using NEON-based special instructions of ARMv8,” *Proceedings of the NIST 3rd PQC Standardization Conference (NIST PQC 2021)*. 2021.

[14] A.Bass, J.M.B.Mera, J.D’Anvers, A.Karmakar, S.S.Roy, M.V.Beirendonck, and F.Vercauteren, “SABER: Mod-LWR based KEM (Round 3 Submission),” *NIST PQC Round 3 submission*, Oct, 2020.

[15]. C.M.M.Chung, V.Hwang, M.J.Kannwischer, G.Seiler, C.J.Shih, and B.Y.Yang, “NTT multiplication for NTT-unfriendly rings new speed records for Saber and NTRU on Cortex-M4 and AVX2,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 159-188, 2021.