64-bit ARMv8 프로세서 상에서의 양자내성암호 최적구현 동향

권혁동* 심민주** 엄시우** 서화정***

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

Optimized implementation trend of post quantum cryptography on 64-bit ARMv8 processor

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

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

요 약

미국 표준과학기술연구소(NIST)에서는 양자컴퓨터 시대에 안전하게 사용할 수 있는 암호 환경을 조성하기 위해 양자내성암호 공모전을 개최하였다. 2022년에 공개키 암호화 부문의 CRYSTALS-Kyber와 전자서명 부문의 CRYSATALS-Dilithium, FALCON, SPHINCS+가 표준으로 선정되었다. 이후 Round 4에서는 공개키 암호화 부문에서 추가적인 표준을 선정하기 위한 공모전이 진행중에 있다. 본 논문에서는 64-bit ARMv8 프로세서를 대상으로 양자내성암호 공모전의 후보들에 대한 최적구현 동향에 대해서 알아본다.

I. 서론

양자컴퓨터 기술의 발전으로 양자 알고리즘인 그루버 알고리즘[1]과 쇼어 알고리즘[2]의 실현이 가능하게 되었다. 이 알고리즘들은 각각 검색과 소인수분해에 효과적이며, 대칭키 암호화 공개키 암호에 치명적이다. 이에 미국 표준과학기술연구소(National Institute of Standards and Technology, NIST)는 양자내성암호 표준화를 위한 공모전을 개최하였다. 본 논문에서는 양자내성암호 표준화 공모전에 제출된 후보 알고리즘들의 64-bit ARMv8 프로세서 상에서의 최적 구현에 대해 알아본다.

II. 배경

2.1 양자내성암호 공모전

양자내성암호 공모전은 2017년에 시작되어 2022년에는 표준 알고리즘이 선정되었다. 공개키 부문에서는 CRYSTALS-Kyber[3]가 선정되었고 전자서명 부문에서는 CRYSTALS-Dilithium[4], FALCON[5], SPHINCS+[6]가 선정되었다. NIST에서는 공개키 부문 알고리즘에 추가 표준을 선정하기 위해서 Round 4를 진행하게 되었다. 이외에 탈락한 알고리즘으로는 공개키 알고리즘인 Saber[7], 전자서명 부문의 FrodoKEM[8], Rainbow[9]가 있다.

2.2 64-bit ARMv8 프로세서

ARMv8 프로세서는 고성능 프로세서로 사물인터넷 기기에서 사용되기도 하지만, 스마트폰과 태블릿 PC에서도 사용되는 프로세서이다[10]. ARMv8 프로세서에는 범용 레지스터 외에도 벡터 레지스터가 존재한다. 벡터 레지스터는 64-bit가 아닌 최대 128-bit까지 저장할 수 있는 대용량이 레지스터이다. 하지만 내부 데이터를 128-bit로 취급하지 않고 일정 단위(8, 16, 32, 64-bit)의 복수 값으로 취급한다. 이런 특징으로 벡터 레지스터는 병렬 연산을 지원한다.

III. 최적 구현물

3.1 FrodoKEM 최적 구현물

ARMed Frodo는 2021년 WISA에서 발표된 암호로 벡터 레지스터를 통한 행렬 곱셈 최적화 기법을 제시하였다. 제안하는 기법은 FrodoKEM-640을 대상으로 행렬 곱 A*s와 s*A를 최적화하며, 추가로 AES 가속기를 적용한 구현물을 제시하였다.

FrodoKEM-640의 행렬 크기는 난수 행렬 A는 640*640, 에러 행렬 s는 640*8 또는 8*640 형태를 지닌다. 행렬의 크기가 매우 큰 관계로 곱셈 연산에 많은 시간이 소요된다. ARMed Frodo는 행렬 곱 연산에서 중복되는 값이 사용되는 부분을 찾아서 해당 값을 벡터 레지스터에 상주시킨다. 그리고 해당 값과 곱해지는 값들을 다른 벡터 레지스터에 호출하여 곱셈을 연산하고 결과 값을 누적하는 형식으로 행렬 곱셈을 완성한다. FrodoKEM-640의 변수는 16-bit 형태이므로 하나의 벡터 레지스터에는 8개의 변수를 저장할 수 있다. 즉, 한번 연산에 8개의 연산 결과 값을 생성한다.

표 1은 ARMed Frodo와 기존 FrodoKEM의 행렬 곱셈기의 성능을 비교한 것이다. ARMed Frodo의 성능이 A*s에서 27.9배, s*A에서 43.8배 빠른 것을 알 수 있다. 해당 곱셈기를 FrodoKEM에 적용하고 AES 가속기를 사용할 경우, 최대 10.22배의 성능 향상이 있었다.

Table 1. Performance comparison of Matrix multiplication algorithms (unit: ms).

Matrix type	A * s	s * A
FrodoKEM[8]	2228.5	2557.7
ARMed Frodo[11]	80.0	58.4

3.2 Rainbow 최적 구현물

Look-up the Rainbow는 2021년에 제안된 기법으로, 사전 연산 테이블을 사용하여 빠른 속도의 곱셈 연산기를 제안하였다[12]. 제안하는 기법은 Tower-field 특성을 사용하여 4-bit*4-bit 곱셈을 미리 계산하여 256-byte의 테이블을 작성하였다. 곱셈 시에는 변수와 상수 값의 곱이라는 점에 착안하여, 실제 테이블 로드는 상수 값에 따라 한 종류 곱셈인 16-byte만 로드하게 설계되었다. Rainbow III와 V에 대해서는 중간에 제곱을 연산할 수 있도록 16-byte의 추가 테이블을 사용하여 총 272-byte의 테이블을 사용한다. 표 2는 제안하는 기법의 곱셈기 성능을 비교한 것이다. 제안하는 기법의 곱셈기가 최대 167.2배 빠른 것을 확인할 수 있다. 해당 곱셈기를 사용할 경우, Rainbow의 성능이 기존 대비 최대 51.6배 향상되었다.

Table 2. Evaluation of multiplier for Rainbow signature (unit: clock cycles).

Multiplier	F_{16}	F_{256}
Rainbow[9]	355	16,557
LUT[11]	58	99

3.3 Scabbard 최적 구현물

Scabbard는 2021년 CHES에서 발표된 Saber 암호의 개선된 형태의 암호이다[13]. Scabbard는 최적 구현한 ARMing-sword는 2022년 WISA에서 발표되었다[14]. ARMing-sword는 Direct Mapping과 Sliding Window 기법을 사용하였다. Direct Mapping은 곱셈 단계 중, 값을 정렬하는 단계에서 중간 단계를 생략하고 바로 결과 값으로 정렬 시키는 기법이다. Sliding Window는 결과 값 누적 시 포인터를 1-byte씩 이동시켜 효과적으로 누적을 진행하는 기법이다. 표 3은 제안하는 기법의 곱셈기를 기존과 비교한 것이다. 제안하는 기법의 곱셈기가 최대 6.34배 빠르다. 이를 사용한 알고리즘의 경우, 최대 2.17배 성능 향상이 있었다.

Table 3. Performance comparison results of ARMing-sword multiplier (unit: clock cycles).

Algorithm	Scabbard[13]	ARMing-sword[14]
EV single	272	137.6
EV 3-way	1,740.8	329.6
EV 4-way	588.8	92.8
Mul. Espada	29.286.4	8,736
Mul. Others	496	425.6

IV. 결론

본 논문에서는 양자내성암호 알고리즘의 ARMv8 상에서의 최적 구현물에 대해 확인하였다. 소개된 알고리즘은 공모전에서 표준으로 선정되지는 못했지만, 연구 또는 학습용으로 사용할 수 있다.

V. Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2018-0-00264, Research on Blockchain Security Technology for IoT Services, 100%).

[참고문헌]

[1] 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.

[2] 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.

[3] 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.

[4] L.Ducas, E.Kiltz, T.Lepoint, V.Lyubashevsky, P.Schwabe, G.Seiler, and D.Stehlé, “Crystals-dilithium: A lattice-based digital signature scheme,” *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238-268, Feb, 2018.

[5] P.A.Fouque, J.Hoffstein, P.Kirchner, V.Lyubashevsky, T.Pornin, T.Prest, T.Ricosset, G.Seiler, W.Whyte, and Z.Zhang, “Falcon: Fast-Fourier lattice-based compact signatures over NTRU,” *Submission to the NIST's post-quantum cryptography standardization process*, Vol. 36, No. 5, 2018.

[6] D.J.Bernstein, C.Dobraunig, M.Eichlseder, S.Fluhrer, S.Gazdag, A.Hülsing, P.Kampanakis, S.Kölbl, T.Lange, M.M. Lauridsen, F.Mendel, R.Niederhagen, C.Rechberger, J.Rijneveld, and P.Schwabe, “SPHINCS,” 2018.

[7] 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.

[8] J.Bos, C.Costello, L.Ducas, I.Mironov, M.Naehrig, V.Nikolaenko, A.Raghunathan, and D.Stebila, “Frodo: Take off the ring! practical, quantum-secure key exchange from LWE,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1006 - 1018, 2016.

[9] J.Ding and D.Schmidt, “Rainbow, a new multivariable polynomial signature scheme,” in *International conference on applied cryptography and network security*, pp. 164 - 175, Springer, 2005.

[10] J.Yiu, “ARMv8-M architecture technical overview,” *ARM white paper*, 2015

[11] H.Kwon, K.Jang, H.Kim, H.Kim, M.Sim, S.Eum, W.K.Lee, and H.Seo, “ARMed Frodo,” In *International Conference on Information Security Applications*, Springer, Cham, pp. 206-217, Aug. 2021.

[12] H.Kwon, H.Kim, M.Sim, W.K.Lee, and H.Seo, “Look-up the Rainbow: Efficient Table-based Parallel Implementation of Rainbow Signature on 64-bit ARMv8 Processors,” *Cryptology ePrint Archive*, 2021.

[13] J.M.B. Mera, A.Karmakar, S.Kundu, and I.Verbauwhede, “Scabbard: a suite of efficient learning with rounding key-encapsulation mechanisms,” *IACR Trans- actions on Cryptographic Hardware and Embedded Systems*, pp. 474 - 509, 2021.

[14] H.D.Kwon, H.J.Kim, M.J.Sim, S.W.Eum, M.W.Lee, W.Lee, and H.J.Seo, “ARMing-sword: Scabbard on ARM,” in proceeding of *The 23rd World Conference on Information Security Applications*(WISA), pp. 23-36, Aug, 2022.