ARMv8에서의 NTT 구현 동향

엄시우*, 심민주*, 송경주*, 서화정**

*한성대학교 (대학원생)
**한성대학교 (교수)

Trends in implementation of Number Theoretic Transform on ARMv8

Si-Woo Eum*, Min-Joo Sim*, Gyeong-Ju Song*, Hwa-Jeong Seo**

*Hansung University(Graduate student)
**Hansung University(Professor)

요 약

NIST PQC 공모전에서 최종 선택된 암호의 대부분은 격자 기반 암호이다. 격자 기반 암호에서는 다항식 곱셈이 많이 사용되며, 일반적으로 정수 다항식 곱셈을 위해 NTT 알고리즘이 사용된다. 격자 기반 암호의 성능 최적화를 위해서는 다항식 곱셈의 최적화가 필수적이다. 본 논문에서는 고성능 임베디드 64-bit 아키텍처인 ARMv8상에서의 NTT 구현 연구 동향에 대해서 알아본다.

I. 서론

최근 NIST PQC의 최종 결과가 발표되었다. 최종 선택된 암호 4개중 3개의 암호는 격자 기반 암호이다. 격자 기반 암호에서는 다항식 곱셈이 많이 사용되며, 성능 향상을 위해서는 다항식 곱셈의 최적화가 필수적이다. 일반적으로 정수 다항식 곱셈을 위해 NTT 알고리즘이 사용되고 있다. ARMv8 기반 프로세서는 고성능 프로세서로 자율 주행 자동차, 다양한 IoT 장치 등 널리 사용되어 오고 있다. 본 논문에서는 ARMv8상에서 NTT 최적화 구현 동향에 대해서 알아본다.

II. 관련 연구

2.1 Number Theoretic Transform

Number Theoretic Transform(NTT)는 계산 복잡도 O(n*logn)를 갖는 다항식 곱셈 알고리즘의 기초로 사용되는 변환이다[ntt]. NTT의 정의는 다음과 같다. 다항식 A(x) 링 R_q에 닫혀 있으며, Vector a는 길이가 n인 벡터로 이루어져 있다. 즉, Vector a = (a_0, …, a_{n-1}) 다항식 A(x)의 계수이다. 이때의 NTT(a)는 수식 (1)로 정의된다.

$$A_i = \sum_{j=0}^{n-1} a_j W_n^{ij} \bmod q, \ (\text{for } i = 0, 1, \cdots, n-1) \quad (1)$$

이때 역변환인 NTT^{-1}(a)는 수식 (2)로 정의된다.

$$A_i = n^{-1} \sum_{j=0}^{n-1} a_j W_n^{-ij} \bmod q, \quad (\text{for } i = 0, 1, \cdots, n-1) \quad (2)$$

NTT^{-1}에서 n-1은 1을 n번 더한 값의 곱셈에 대한 역원이다. 이때 NTT 기반 다항식 곱셈 연산은 Convolution 정리를 적용하여 계산 가능하다. Convolution 정리와 NTT를 사용한 다항식

곱셈기의 수식은 다음과 같다.

$$a \times b = NTT^{-1}(NTT(a) \circ NTT(b))$$

다항식 a,b를 NTT 과정을 수행하고 결과값을 곱셈이 아닌 Convolution을 사용한다. 그 다음 역 변환 NTT^{-1}을 수행하면 다항식 곱셈의 결과가 나온다. 이를 통해 일반적인 다항식 곱셈의 계산 복잡도를 $O(n^2)$에서 O(n*logn)로 줄일 수 있다[1, 2].

2.2 ARMv8 Architecture

ARMv8 아키텍처는 64-bit(AArch64), 32-bit(AArch32) 아키텍처 모두를 지원하는 ARM의 고성능 임베디드 64-bit 아키텍처이다. 64-bit로 사용 가능한 x0-x30의 범용 레지스터 31개를 제공하고, 이 범용 레지스터는 w0-w30으로 32-bit 단위로도 사용 가능하다. 벡터 레지스터는 128-bit 크기를 가지며, v0-v31로 32개를 제공하고 있다[3].

III. ARMv8에서의 FFT 구현

3.1 Neon NTT: Faster Dilithium, Kyber, and Saber[4]

2021년 Becker et al.은 NIST PQC 격자 기반 암호(Dilithium, Kyber, Saber)들의 다항식 곱셈의 중심 기술인 FFT의 정수 아날로그인 NTT에 대해 연구하였다. NTT외의 다항식 곱셈에 대한 다른 곱셈기(Toom-Cook, Karatsuba)와 비교도 같이 진행하였다.

효율적인 곱셈을 위하여 몽고메리 곱셈과 거의 동일하지만 NEON에 특히 적합한 Barrett-reduction 기반의 알려진 인자 곱셈을 포함하여 기존 구현을 개선한 새로운 구현 기법을 제안하였다. 몽고메리 곱셈과, Barrett-reduction, 다단계의 버터플라이 결합은 훨씬 빠른 성능을 보여주었다. 또한 불완전한 NTT의 결과(Saber,Kyber)를 캐싱할 때마다 적용 할 수 있는 “비대칭 기본 곱셈”을 제안하였다.

결과적으로 Saber에서 NTT는 Toom-Cook 곱셈보다 2배 높은 성능을 보여주었다. 또한 Apple M1에서 Kyber와 Saber의 구현시 19~36% 빠른 성능을 보여주었다.

3.2 Accelerating Falcon on ARMv8[5]

2022년 Kim et al.d은 디지털 서명 알고리즘은 Falcon의 FFT/NTT 기반 다항식 곱셈 최적화를 진행하였다. Falcon은 NIST에서 진행하고 있는 PQC(Post-Quantum Cryptography) 표준화에서 최종 선택된 디지털 서명 알고리즘이다. Falcon의 주요 계산은 복소수 영역과 정수 영역의 다항식 곱셈이다. 일반적으로 NTT 기반의 곱셈방식은 정수 영역의 효율적인 다항식 곱셈을 위해 사용되어 왔다.

Falcon에서 키 생성, 서명, 검증에서는 여러 NTT 알고리즘이 사용된다. Falcon은 다른 PQC 디지털 서명 알고리즘에 비해 서명 검증 속도가 빠른 장점이 있으며, 이 이점을 더욱 강조하기 위한 병렬 구현에 중점을 두며, NEON 벡터 레지스터의 크기와 Falcon의 모듈러스를 고려하여 8개의 계수를 동시에 수행하는 CT-Butterfly 병렬 구현 기법을 제안하였다. CT-Butterfly 연산은 NTT의 핵심 연산이며 제안하고 잇는 병렬 구현은 직렬 구현 대비 여러 연산을 동시에 처리하므로 더 빠른 성능을 보여준다. 또한 NEON 엔진에서 사용 가능한 레지스터를 최대한 활용하여 중복 메모리 접근을 최소화하였다.

결과적으로 Falcon의 키 생성에서 15.1%, 16.5%, 65.4%의 성능 향상을 보여주었다.

3.3 Crystals-Dilithium on ARMv8[6]

2022년 Kim et al.은 Dilithuum의 성능 향상을 위하여 큰 레지스터 세트 및 NEON 엔진과 같은 ARMv8의 아키텍처 속성을 활용하여 핵심 연산인 NTT 기반 다항식 곱셈을 최적화하였다.

Crystals-Dilithium은 NIST PQC의 최종 디지털 서명 알고리즘 중 하나이다. Dilithium은 LWE(Learning With Error) 문제의 경도를 사용하는 격자 기반 알고리즘이다. 격자 기반 알고리즘에서는 NTT 기반 다항식 곱셈 알고리즘의 최적화가 필수적이다.

NTT 곱셈은 NTT, NTT^{-1}, Point-wise 곱셈으

로 구분되기 때문에 각각의 구체적인 최적환 전략을 소개하고 있으며, ARM/NEON 프로세서를 둘다 사용한 통합 버터플라이 방식의 Interleaving 개념을 제안하였다. NTT 곱셈에서 가장 계산 비용이 많이 드는 NTT와 NTT^{-1}에서 메모리 접근을 최소화 하고 NEON 엔진을 사용하여 효율적인 병렬 구현을 제안하였다. 결과적으로 통합 및 레지스터 유지 방식을 통해 메모리 접근을 최소화 할 뿐만 아니라 ARM 프로세서에서 일부 계수의 NTT를 처리하여 ARM 연산의 대기시간을 NEON 오버헤드로 숨김으로써 성능을 향상시켰다.

결과적으로 제안하는 기법들을 적용하여 NIST PQC 최종 라운드에 제출된 참조 구현과 비교하였을 때, 키 생성에서는 약 43%, 서명에서는 약 113%, 검증에서는 약 41% 성능 향상을 달성하였다.

IV. 결론

본 논문에서는 ARMv8상에서의 NTT 최적화 구현 동향에 대해 알아보았다. 다가오는 양자컴퓨터 사용에 대비하여 개발되고 있는 PQC 중 격자 기반 암호에서는 NTT 기반 다항식 곱셈이 많이 활용되고 있으며, 속도 향상을 위해 NTT 최적화가 필수가 되어가고 있다. 특정 암호에서는 NTT 외의 다른 곱셈기를 활용할 경우 더 효율적인 결과도 보여주고 있기 때문에 다양한 곱셈기와 비교한 연구가 필요하다.

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] 김동현. "NTT 기반 다항식 곱셈기를 이용한 고성능 Ring-LWE 암호시스템 설계." 국내석사학위논문 인하대학교 대학원, 2018.

[2] 이제원, 김민지, 김영효, 전창열, 김동찬. “다항식 몫환 Fp[X]/<Xn+1>의 NTT 기반 곱셈 연산에 관한 연구.” 한국통신학회 학술대회논문집. pp. 1730-1731. Jun, 2022.

[3] Armv8-A Instruction Set Architecture. [online] available: https://documentation-service.arm.com/static/613a2c38674a052ae36ca307?token=.

[4] Becker, Hanno, et al. "Neon NTT: faster dilithium, kyber, and saber on cortex-a72 and apple M1." Cryptology ePrint Archive. 2021.

[5] Kim, Youngbeom, Jingyo Song, and Seog Chung Seo. "Accelerating Falcon on ARMv8." IEEE Access 10 : 44446-44460. 2022.

[6] Kim, Youngbeom, et al. "Crystals-Dilithium on ARMv8." Security and Communication Networks 2022 (2022).