

Cortex-M4 상 NIST PQC 서명 스킴 PERK

실행 시간 분석 및 최적화 방안

이수성, 김영식*

대구경북과학기술원 전기전자컴퓨터공학과

mercury@dgist.ac.kr, ysk@dgist.ac.kr

Execution Time Analysis and Optimization Strategies for

the NIST PQC Signature Scheme PERK on Cortex-M4

Su-Seong Lee, Young-Sik Kim*

Department of Electrical Engineering and Computer Science, DGIST

요 약

본 논문에서는 NIST PQC 추가 전자서명 후보군 2 라운드에 선정된 PERK 전자서명 알고리즘을 대상으로, 임베디드 환경에서의 실행 성능을 분석하였다. PERK 알고리즘의 키 생성, 서명, 검증 각 단계의 실행 시간은 ARM Cortex-M4F 코어를 탑재한 STM32F407 마이크로 컨트롤러에서 측정되었으며, 측정은 STM32CubeIDE의 Serial Wire Viewer(SWV) 기능을 이용하여 수행하였다. 실험 결과, 전체 실행 시간의 약 70%가 KeccakF1600StatePermute 함수에서 소요되었고, int32_sort 함수는 약 9%의 비중을 차지하였다. 두 연산 모두 병렬화에 적합한 구조를 가지므로, FPGA와 같은 병렬 하드웨어 환경에서 구현할 경우 실행 시간을 효과적으로 단축할 수 있을 것으로 기대된다.

I. 서 론

양자 내성 암호 (Post-Quantum Cryptography, PQC)는 기존 암호에 비해 연산량과 메모리 요구량이 크다. 이로 인해 임베디드 시스템이나 저사양 IoT 디바이스에서 성능 저하를 초래할 수 있어, 제한된 자원 환경에서의 효율적 구현을 위한 최적화 연구가 필수적이다. 이를 위해서는 알고리즘 내에서 병목이 발생하는 연산에 대한 분석이 선행되어야 한다.

분석 대상으로는 PERK를 선정하였다. PERK는 Permuted Kernel Problem (PKP)의 어려움에 기반하는 전자서명 알고리즘으로, NIST PQC 추가 전자서명 알고리즘 2 라운드에 진출한 후보 중 하나다. short-3 파라미터를 사용할 경우 6.25KB의 서명 크기, 0.15KB의 공개키, 16B의 비밀키를 가지기 때문에, 임베디드 환경에 적합하다. [1]

본 논문에서는 PERK의 키 생성, 서명, 검증 각 단계의 실행 시간을 ARM Cortex-M4F 코어를 탑재한 STM32F407 마이크로 컨트롤러에서 측정하고, 병목 지점을 분석한 후, 이를 바탕으로 향후 최적화 방향을 제안한다.

II. 본론

본 논문에서는 pqm4 프로젝트에서 제공하는 PERK 알고리즘 구현[2]을 ARM Cortex-M4F 기반의 STM32F407 마이크로 컨트롤러 환경에 맞게 수정하여 측정하였다. 해당 마이크로 컨트롤러는 168MHz의 클럭 속도, 128KB의 RAM, 1MB의 플래시 메모리를 탑재하고 있는 칩으로, 제한된 자원 환경에서 PERK 알고리즘의 성능을 평가하기에 적절하다.

실행 시간은 STM32CubeIDE에서 제공하는 Serial Wire Viewer(SWV) 기능을 이용하여 측정하였다. SWV는 약 7168 클럭 사이클마다 현재 실행 중인

함수의 주소를 기록하는 방식으로 동작하며, 이를 기반으로 각 함수가 전체 실행 시간 중 차지하는 비중을 누적하여 분석했다. 측정은 10 회 반복하였으며, 측정된 실행 시간 데이터를 바탕으로 PERK 알고리즘 내에서 성능 병목이 발생하는 부분을 식별하고 향후 최적화 방향을 도출했다.

III. 결론

PERK128_fast3 과 PERK128_short3 을 측정하여 각 단계에서 가장 비율이 높은 세개의 함수를 표시한 결과는 각각 표 1 과 표 2 와 같다. KeccakF1600 이 PERK128_fast3 과 PERK128_short3 에서 각각 실행 시간의 45.7%, 42.5%를 차지하였으며, int32_sort 는 각각 28.7%, 27.5%를 차지하는 것으로 나타났다. KeccakF1600 는 SHA-3 의 block permutation 을 구현한 것이다. int32_sort 는 부채널 공격을 방지하기 위해 상수 시간(Constant-time) 정렬 알고리즘인 Bitonic Sort 를 사용하였다.

KeccakF1600 과 Bitonic Sort 는 모두 병렬화에 적합한 구조를 가지고 있고, 본 구현에서는 마이크로 컨트롤러의 한계로 병렬연산을 지원하지 않았기 때문에 FPGA 환경에서 병렬 구현을 적용할 경우 전체 실행 시간을 효과적으로 단축시킬 수 있을 것으로 기대된다.

PERK128 _fast3	keygen	0.29%	KeccakF1600_	36.55%
			sig_perk_mat_vect_mul	25.78%
			sig_perk_mat_set_random	18.77%
	sign	69.25%	KeccakF1600_	44.04%
			int32_sort	29.85%
			uint32_sort	3.10%
	verify	30.45%	KeccakF1600_	49.69%
			int32_sort	26.41%
			sig_perk_perm_gen_given_random_input	4.33

표 1. PERK128_fast3 전자서명 알고리즘의

주요 함수별 실행 시간 비율

PERK128 _short3	keygen	0.05%	KeccakF1600_	36.29%
			sig_perk_mat_vect_mul	26.30%
			sig_perk_mat_set_random	18.38 %
	sign	67.60%	KeccakF1600_	41.59%
			int32_sort	29.06%
			uint32_sort	3.00%
	verify	32.35%	KeccakF1600_	44.45%
			int32_sort	24.17%
			sig_perk_perm_gen_given_random_input	3.87%

표 2. PERK128_short3 전자서명 알고리즘의

주요 함수별 실행 시간 비율

ACKNOWLEDGMENT

이 논문은 2024 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(RS-2024-00399401, 양자안전 보안인프라 전환 및 대양자 복합 안전성 검증기술 개발).

참 고 문 헌

[1] N. Aaraj, S. Bettaieb, L. Bidoux, A. Budroni, V. Dyseryn, A. Esser, T. Feneuil, P. Gaborit, M. Kulkarni, V. Mateu, M. Palumbi, L. Perin, M. Rivain, J.-P. Tillich, and K. Xagawa, "PERK Specification (v2023.11)," 2023, (<https://pqc-PERK.org/resources.html>)

[2] D. Müller, T. Pöppelmann, P. Schwabe, F. Virdia, and R. Avanzi, "pqm4: Post-quantum cryptography on the ARM Cortex-M4," 2024. (<https://github.com/mupq/pqm4>)