Toward Efficient and Fair Software/Hardware Codesign and Benchmarking of Candidates in Round 2 of the NIST PQC Standardization Process

Farnoud Farahmand, Viet B. Dang, Michał Andrzejczak*, Duc Tri Nguyen, and Kris Gaj
George Mason University, U.S.A.
*on leave from the Military University of Technology in Warsaw, Poland

Post-Quantum Cryptography (PQC) refers to a new class of cryptographic algorithms that are resistant against all known attacks using quantum computers, but at the same time can be implemented by themselves using traditional computing platforms, such as microprocessors, microcontrollers, Field Programmable Gate Arrays (FPGAs), and Application Specific Integrated Circuits (ASICs). PQC is a cryptographic community's response to the emerging threat of full-scale quantum computers, expected to be developed within the next decade or two. The main goal of PQC is to replace the existing public-key cryptography standards, based on RSA and Elliptic Curve Cryptography, which seem to be the most vulnerable to quantum computing and impossible to defend using traditional approaches, such as gradually increasing key sizes.

In order to initiate a timely transition to a new class of cryptographic schemes, in December 2016, NIST issued an official "Call for Proposals and Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms," commencing the NIST PQC standardization process. The number of submissions qualified to Round 1, started in December 2017, reached 69. In January 2019, based on the results of the initial security analysis and preliminary software benchmarking, 26 submissions were qualified by NIST to Round 2. These submissions included multiple public-key encryption, key encapsulation mechanism (KEM), and digital signature schemes, often more than one per a single submission.

Traditionally, hardware benchmarking played a major role in all recent cryptographic standardization efforts, such as AES, eSTREAM, SHA-3, and CAESAR contests. Unfortunately, this trend is not likely to be sustained in case of the NIST PQC standardization process, by simply following the old practices and hardware benchmarking approaches. In many respects, PQC schemes are dramatically different from those evaluated in previous cryptographic contests, and new challenges call for new substantially different solutions.

During the past contests, software and hardware benchmarking were conducted separately, by different groups of experts, equipped with different knowledge and tools. For PQC algorithms, this approach is hard to maintain. These algorithms are simply too complex and too different from the current state-of-the-art to permit the development of optimized purely hardware implementations of a significant percentage of the remaining candidates by a single group within the time frame imposed by the NIST evaluation process (about 12-18 months per single round).

At the same time, there is little if any consensus, regarding basic design choices. In 16 months since the start of the PQC Round 1 (or before), only a few purely hardware implementations of Round 1 candidates were announced and even fewer were made open source. These implementations used different Application Programming Interfaces (APIs), targeted different platforms, and aimed at different optimization targets from high-speed to low-area. No conclusions regarding ranking of these algorithms in terms of their performance in hardware can be reached based on such divergent efforts.

In this talk, we propose a new approach to systematic benchmarking of candidates in cryptographic contests, based on the development and experimental measurements of their software/hardware codesigns. This approach is particularly applicable to the current stage of the NIST PQC standardization process, where a large number and high complexity of the evaluated algorithms makes the traditional

hardware benchmarking practically infeasible. We propose and justify the choice of a suitable platform and design methodology. We demonstrate the validity of our approach by applying it to 7 Key Encapsulation Mechanisms (KEMs), representing 5 NIST Round 2 PQC candidates.

The obtained results indicate a potential for very substantial speed-ups vs. purely software implementations, ranging between 5 and 187 for encapsulation and between 15 and 444 for decapsulation. These speed-ups depend primarily on the percentage of the software execution time taken by functions offloaded to hardware (rather than the amount of acceleration itself). Ranking of the investigated candidates is affected, but not dramatically changed, by hardware acceleration.

At the same time, it should be noted that our current study cannot be used to predict the performance and ranking of the investigated candidates when implemented entirely in hardware. Such implementations can further benefit from elimination of the communication overhead between a processor and a hardware accelerator. They may also take advantage of an ability to parallelize some additional operations, left in software in the current study. As a result, more effort, by multiple groups, is needed to determine and realize the most efficient and fair software/hardware partitioning schemes, and to extend our study to the remaining Round 2 PQC candidates.