Post-Quantum Cryptography in Reconfigurable Hardware: Challenges, Opportunities, and State-of-the-Art

Kris Gaj, Ahmed Ferozpuri, Viet Dang, Duc Nguyen,
Farnoud Farahmand, and Jens-Peter Kaps
George Mason University
U.S.A.

Practical Quantum Computers have been recently selected as one of ten breakthrough technologies of 2017 by MIT Technology Review. Major investment by companies, such as Google, IBM, Intel, Microsoft, and NTT led to the first general-purpose quantum processors, with up to 72 physical qubits, and specialized processors reaching 2048 qubits. In parallel, significant progress has been made in the area of languages for describing quantum computing algorithms, software development kits, and quantum computing simulators. An era of so-called quantum supremacy, when quantum computers can solve problems beyond the reach of practical computers, seems to be on the horizon, with remaining difficulties mostly in the realm of engineering and technology.

When established as a mature technology, quantum computers would easily break all current American federal and banking standards in the area of public-key cryptography, including algorithms protecting the majority of the Internet traffic, such as RSA, Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA), and Diffie-Hellman. All traditional methods of dealing with growing computational capabilities of potential attackers, such as increasing key sizes, would be futile.

As a result of this potential threat, a new field of Post-Quantum Cryptography (PQC) has emerged. PQC is devoted to the design and analysis of cryptographic algorithms that are resistant against any known attacks using quantum computers, but by themselves can be implemented using classical computing platforms, such as general-purpose microprocessors, microcontrollers, ASICs and FPGAs.

The scientists pursuing PQC have established a series of conferences, called PQCrypto, held regularly since 2006. In Feb. 2016, American National Institute of Standards and Technology (NIST) announced its plans of starting the standardization effort in the area of PQC. This effort is predicted to last several years and result in an entire portfolio of algorithms capable of replacing current public-key cryptography schemes. The initial announcement was followed by the official "Call for Proposals and Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms," issued in Dec. 2016. Nov. 30, 2017 was set as the candidate submission deadline. After the initial review by the NIST staff, it appeared that 69 distinct algorithms (each with multiple parameter sets corresponding to up to 5 different security strengths) were recognized as valid submissions, and their corresponding specifications and software implementations posted on the NIST website. None of the 69 submissions for the future NIST PQC standard included a corresponding hardware implementation.

After publishing the submission packages on the NIST website on Dec. 22, 2017, the evaluation process started almost immediately. The evaluation is likely to last between three and five years, and be similar to that conducted in the case of earlier NIST cryptographic standards, such as Advanced Encryption Standard (AES) and Secure Hash Algorithm 3 (SHA-3). In each of these cases, the number of candidates was gradually narrowed down, in consecutive rounds, to those with adequate security (typically less than 1/3rd of the initial pool), and the final selection was based on software and hardware efficiency. In both of the aforementioned competitions, the candidate most efficient in hardware was announced the winner.

In this talk, we will provide the motivation for the evaluation of the PQC candidates in hardware. Five major families of PQC schemes, code-, hash-, isogeny-, lattice-, and multivariate-based, will be shortly introduced, together with their major representatives, and hardware implementation-oriented characteristics. For each family, their support for signature, encryption, and Key Encapsulation Mechanism (KEM) schemes will be explored.

Implementation and benchmarking of PQC algorithms in hardware involves overcoming a large range of unique challenges. The biggest of them is the complexity of the algorithms and their often highly mathematical specifications. As a result, the workload for even a single algorithm, implemented using the traditional RTL approach, can easily reach several man-months. Close collaboration with cryptographers responsible for designing a particular algorithm is highly recommended and often indispensable.

The High-Level Synthesis (HLS) approach appears to be very appealing (especially in light of a very large number of Round 1 PQC candidates and the availability of reference and optimized implementations in C). However, to our knowledge, this approach has not been yet reported for any PQC algorithm, not to mention contrasted, with the RTL approach in terms of performance and/or resource utilization. If attempted, it may lead to potentially unfair comparisons, due to the lack of any clear indication when the optimization of the HLS-ready C code should end and the limited experience of both hardware designers and software programmers with this new methodology.

The second challenge is the storage of large public keys, private keys, and internal state, inside of a hardware module, which may prohibit truly lightweight implementations and effect the key agility of all hardware implementations. The other difficulty is the requirement for random numbers, used as inputs for encryption, signature generation, and key encapsulation with the specific (e.g., the uniform or Gaussian) distribution.

The majority of PQC algorithms do not use the same basic operations as classical public-key cryptosystems, not to mention secret-key cryptosystems. As a result, the corresponding building blocks may need to be developed from scratch and thoroughly optimized. If these development efforts are undertaken independently by multiple designers, that may easily lead to unfair and biased comparisons.

Unlike it is the case for secret-key algorithms, the majority of countermeasures against side-channel attacks are algorithm-specific, and yet unexplored for the majority of Round 1 PQC candidates. Their development and experimental validation is likely to require a very considerable and prolonged effort, including substantial modifications and extensions of experimental frameworks.

The first step in addressing the aforementioned challenges has been the development of the PQC Hardware API, proposed by the GMU Team. This API includes the minimum compliance criteria aimed at the fair evaluation of all developed implementations. For example, the key generation is required to be implemented outside of a PQC module (either in software or in a separate hardware module) as the operations required for this functionality are often substantially different from those used for encryption/decryption, signature generation/verification, and key encapsulation/decapsulation. On the other hand, the two basic functionalities (e.g., encryption and decryption) are required to be always implemented together to demonstrate the algorithm's potential for resource sharing.

The next step will be the creation of the Development Package, containing HDL code that can be shared among multiple PQC candidates (including any pre-processing and post-processing units, libraries of basic operations, and universal testbenches). The experiments with contrasting the results of the corresponding RTL and HLS implementations will follow, in parallel with an effort to develop and validate both universal and algorithm-specific countermeasures against side-channel attacks and fault attacks.

The undertaken effort will greatly benefit from the pilot studies, concerning the implementations of earlier versions of the PQC algorithms, conducted by several cryptographic engineering research groups worldwide, including the GMU group. In this talk, we will review and summarize these early results, and highlight the insights they provide regarding the potential hardware performance characteristics of various PQC families. At the same time, the results of these earlier studies will need to be used with considerable caution, due to the recent changes in the functionality and parameter values of even well-established candidates, such as NTRUEncrypt, Rainbow, Unbalanced Oil-and-Vinegar, etc. Additionally, the previous implementations were using very divergent assumptions, optimization targets, APIs, and sources of randomness, which makes their direct use for comparative analysis challenging.

Thus, this talk will be aimed at encouraging and facilitating the involvement of the CryptArchi community in the PQC standardization efforts, and at connecting members of this community with the NIST submission teams. By getting involved in the reconfigurable hardware implementation and benchmarking of PQC algorithms, the CryptArchi researchers will have a unique opportunity to influence the choice of future cryptographic standards, that are likely to be developed and deployed within the next decade and remain in use for the significant portion (if not the rest) of the 21st century.