Lessons Learned from High-Speed Implementation and Benchmarking of Two Post-Quantum Public-Key Cryptosystems

Malik Umar Sharif, Ahmed Ferozpuri, and Kris Gaj
George Mason University

Abstract

If a quantum computer with a sufficient number of qubits was ever built, it would easily break all current American federal 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.

In Feb. 2016, American National Institute of Standards and Technology (NIST) has published a draft report and announced its plans of starting the standardization effort in the area of post-quantum cryptography. This effort is likely to last years and result in the 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. As a part of this standardization process, fair and efficient benchmarking of Post-Quantum Cryptography (PQC) algorithms in hardware and software becomes a necessity.

In this talk, we will discuss our hardware high-speed implementations of two PQC schemes:

1. NTRUEncrypt Short Vector Encryption Scheme (SVES), fully compliant with the IEEE 1363.1 Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices, and
2. Multivariate Rainbow Signature Scheme.

For fair comparison, both implementations follow the same PQC Hardware Application Programming Interface (API), proposed by our group. The development process, was also standardized, and included common intermediate deliverables, such as
a) detailed flow diagrams, b) choice of supported parameter sets, c) top-level block diagram, d) lower-level block diagrams, e) parameters of a hardware architecture, f) cycle-based timing analysis, g) Algorithmic State Machine (ASM) charts, h) Register-Transfer Level (RTL) code, i) software-generated test vectors, j) comprehensive testbenches, k) results of synthesis and implementation, l) analysis of results, m) lessons learned.

As such, both designs provide a valuable reference for any future hardware implementers of PQC schemes, which is very important in the context of the upcoming NIST standard candidate evaluation process.

To the best of our knowledge, we have developed the first synthesizable HDL code of the entire NTRUEncrypt SVES scheme, reported in the scientific literature or available commercially. Our implementation supports two representative parameter sets specified in the 2009 IEEE 1363.1 Standard: ees1087ep1 and ees1499ep1, optimized for speed,

which provide security levels of 192 and 256 bits, respectively. The corresponding public key sizes are 1495B and 2062B, respectively, and the corresponding private key sizes, 87B and 109B.

Our hardware implementation is functionally equivalent to the open source software implementation of the IEEE P1363.1 standard, developed by Security Innovation, Inc., and has been thoroughly verified using test vectors generated using this implementation. The speed up of our hardware design running on Xilinx Virtex-7 XC7VX485T FPGA vs. software implementation, running on the Cortex A9 ARM Core of Zynq 7020, with the clock frequency of 666.7 MHz, is over 400.

The relative contribution of various operations to the total execution time is substantially different for the hardware and software implementations. In software, Polynomial Multiplication amounts to about 90% of the total execution time. On the other hand, our hardware implementation is seriously limited by the sequential nature of the SHA-256 calculations. As a result, the operations that are most critical are hash based operations of the Blinding Polynomial Generation Method (BPGM) and the Mask Generation Function (MGF), amounting to about 83% of the total execution time for both supported parameter sets. At the same time, the Polynomial Multiplication can be almost completely overlapped with the computations of BPGM through the use of pipelining, and thus has a negligible influence on the execution time.

In order to remove the hash function bottleneck, multiple solutions have been proposed, including an unrolled hash function architecture, as well as a replacement of the SHA-2/SHA-1 hash functions by more hardware friendly SHA-3 or a pseudorandom function based on the pipelined implementation of AES.

Using the similar methodology, we have developed a new high-speed hardware implementation of the multivariate Rainbow signature scheme, based on the earlier work by Tang et al., presented at the PQCrypto 2011 conference. Our implementation targets an 80-bit security level, through the choice of the following variant of the Rainbow Signature Scheme: Rainbow (GF(2^8), 17, 12, 12). It has a public key size of 19.64kB and private key size of 30.94kB. It is based on the parallel hardware design for the Gauss-Jordan elimination, which is capable of solving an NxN system of linear equations over GF(2^8) in N clock cycles. In an effort to optimize the design, multiple architectures for the two-input and three-input multipliers over GF(2^8) have been implemented and comprehensively benchmarked. Additionally, a novel hardware architecture for the so called pivoting operation (a part of the Optimized Gauss-Jordan Elimination) has been developed.

Our hardware implementation is functionally equivalent to the software implementation by Jintai Ding and Dieter Schmidt from University of Cincinnati, and has been comprehensively verified using test vectors generated using this implementation. The result generation and analysis is currently in progress, and will be presented at the workshop. As a next step, the current architecture will be extended to support at least one additional parameter set, with the higher security level.

Eventually, our goal is to compare the hardware implementations of NTRUEncrypt SVES and the Rainbow Signature Scheme at the same security level, using the same API, from the point of view of the execution time, resource utilization, and speed-up vs. software, as well as flexibility and scalability in terms of supporting multiple parameter sets. This project is intended to pave the way for the future comprehensive, fair, and efficient hardware benchmarking of the most promising encryption, signature, and key agreement schemes from each of several major post-quantum public-key cryptosystem families.