FPGA-based Post-Quantum Secure Niederreiter Cryptosystem Demonstration

Wen Wang[1], Jakub Szefer[1], and Ruben Niederhagen[2]

[1] Yale University, New Haven, CT, USA
{wen.wang.ww349, jakub.szefer}@yale.edu
[2] Fraunhofer SIT, Darmstadt, Germany
ruben@polycephaly.org

1 Research Description

We will present a demo showing the fastest-to-date FPGA-based implementation of the Niederreiter cryptosystem using binary Goppa codes, including modules for encryption, decryption, and key generation. Our implementation is constant-time in order to protect against timing side-channel analysis. The design is fully parameterized, using code-generation scripts, in order to support a wide range of parameter choices for security, including binary field size, the degree of the Goppa polynomial, and the code length. The parameterized design allows us to choose design parameters for time-area trade-offs in order to support a wide variety of applications ranging from smart cards to server accelerators. For parameters that are considered to provide "128-bit post-quantum security" (i.e., the cost of an attack on a quantum computer is assumed to be at least 2^{128} quantum operations), our time-optimized implementation requires 966,400 cycles for the generation of both public and private portions of a key and 14,291 cycles to decrypt a ciphertext. The time-optimized design uses only 121,806 ALMs (52% of the available logic) and 961 RAM blocks (38% of the available memory), and results in a design that runs at about 250 MHz on a medium-size Stratix V FPGA (5SGXEA7N).

This demonstration is based on our on-going research. The details of the key generator module have been presented in [3]. One of the key modules within the key generator — Gaussian Systemizer, was presented in [2]. Our most recent FPGA-based work involving the full Niederreiter cryptosystem [4] is to appear in PQCrypto this year.

2 Hardware Setup

Experimental Setup. As shown in Figure 1, the proposed hardware setup for the demonstration includes: a Stratix V FPGA (5SGXEA7N), a Linux CPU (host computer) connected to a display, power supply for FPGA, a USB cable, and a RS422 cable. Users can send different commands from the terminal to the FPGA through the serial port e.g., set parameters, send seeds, generate a key pair, encrypt a message, and decrypt a ciphertext. The FPGA will send acknowledgements and return the results to the host computer. All of these commands and acknowledgements will be shown on the display during runtime. Once the FPGA finishes the computations and returns the final result, the host

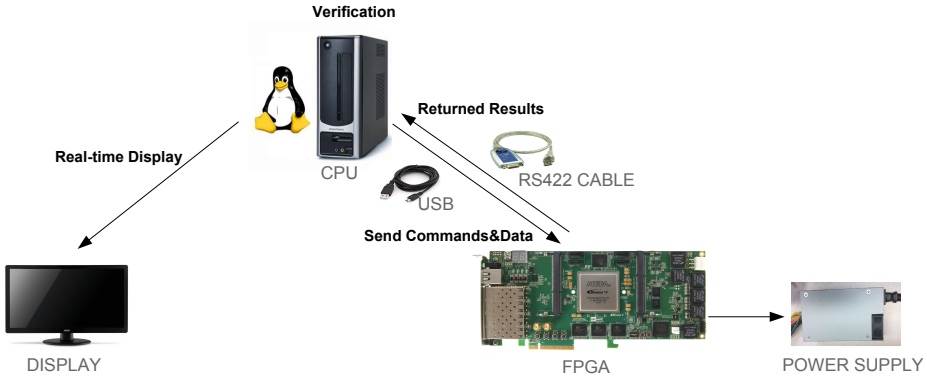

Fig. 1: Hardware Setup Diagram.

computer will check the correctness of the computation by verifying the results with a software reference implementation.

Research Feathers and Audience Expectations. The demonstration will highlight the significant speedup that a hardware (FPGA) implementation can achieve over existing software implementations. Once the FPGA computation is started, the to-date fastest software implementation of the Niederreiter cryptosystem [1] will be started as well. The audience will be able to observe the computation status of the whole Niederreiter cryptosystem (set parameters, send seeds, key generation, encryption and decryption), check the returned acknowledgements from the FPGA, and see how the CPU checks the correctness of the results returned by the FPGA. Interested audiences can learn how to send the commands from the terminal and run the Niederreiter cryptosystem on the FPGA themselves easily since the whole process is fully automatic and straightforward. More importantly, by comparing the performance of our FPGA-based design with the CPU-based design [1], audiences are able to see how FPGA helps to accelerate the computations within post-quantum cryptosystems, despite the over 10x slower clock of the FPGA. The HDL and software source code of this demonstration is available at `http://caslab.csl.yale.edu/code/niederreiter/`.

References

1. Chou, T.: McBits revisited. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 213–231. Springer, Heidelberg (2017)
2. Wang, W., Szefer, J., Niederhagen, R.: Solving large systems of linear equations over GF(2) on FPGAs. In: Reconfigurable Computing and FPGAs – ReConFig 2016. pp. 1–7. IEEE (2016)
3. Wang, W., Szefer, J., Niederhagen, R.: FPGA-based key generator for the Niederreiter cryptosystem using binary Goppa codes. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 253–274. Springer, Heidelberg (2017)
4. Wang, W., Szefer, J., Niederhagen, R.: FPGA-based Niederreiter cryptosystem using binary Goppa codes. In: To appear in the Proceedings of the International Conference on Post-Quantum Cryptography. PQCrypto (April 2018)