Post-Quantum Cryptography on FPGAs: the Niederreiter Cryptosystem

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

[1]Computer Architecture and Security Laboratory, Yale University, USA
[2]Fraunhofer SIT, Darmstadt, Germany

caslab.csl.yale.edu

sit.fraunhofer.de/en

1. Project Overview

Once sufficiently large quantum computers are built, Shor's algorithm can solve the integer-factorization problem and the discrete-logarithm problem in polynomial time, which would allow breaking cryptosystems built upon the hardness assumptions of these problems, e.g., RSA, ECC, and Diffie-Hellman. In addition, Grover's algorithm gives a square-root speedup on search problems and improves brute-force attacks that threatens, for example, symmetric key ciphers like AES.

In our project, we present the first post-quantum secure, constant-time, efficient, and tunable FPGA-based implementation of the Niederreiter cryptosystem using binary Goppa codes.

2. Background

post-quantum cryptography

lattice | **code** | hash | multivariate | isogenies

Binary Goppa code

- degree-t Goppa polynomial $g(x) \in GF(2^m)[x]$
- code locator $L = \{\alpha_0, \ldots, \alpha_{n-1}\}$, $g(\alpha_i) \neq 0$, $\alpha_i \in GF(2^m)$
- can be defined by a parity check matrix H, e.g., $C = \{c \mid Hc = 0\}$

Niederreiter encrypt

- e: error vector of weight t
- syndrome $S = He$

Niederreiter decrypt

- compute e given the syndrome S

3. Contributions

We present a full cryptosystem with tunable parameters, which uses code-generation to generate vendor-neutral Verilog HDL code.

Dedicated hardware implementations of:

- Gaussian systemizer which works for any large-sized matrix over any finite binary field.
- Gao-Mateer Additve FFT for polynomial evaluation.
- Merge sort for obtaining uniformly distributed permutations.
- Constant-time Berlekamp-Massey decoding algorithm.

We test the design using Sage reference code, iVerilog simulation, and output from real FPGA runs.

4. Hardware Design

Figure 1 shows the hardware design dataflow for three main parts:

(a) key generation, (b) encryption, and (c) decryption of the full Niederreiter cryptosystem by use of the dedicated hardware functional units we built.

Dark gray boxes represent block memories, while white boxes represent major logic modules.

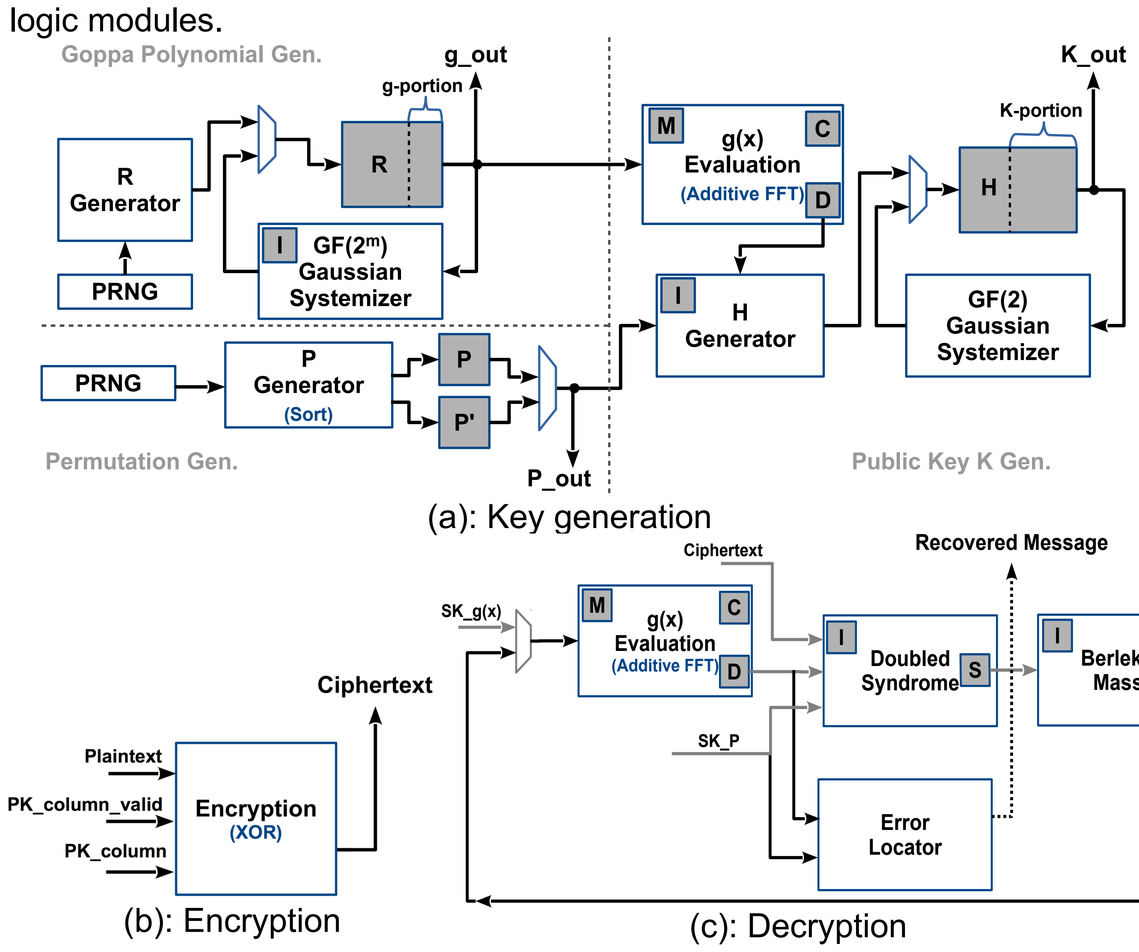

Fig.1: Dataflow diagrams of the full cryptosystem

5. Evaluation Setup

Figure 2 shows the testing setup. We implemented a serial IO interface for communication between the host computer and the FPGA. The interface allows us to send data and simple commands from the host to the FPGA and receive data, e.g., public and private key, ciphertext, and plaintext, from the FPGA. We verified the correct operation of our design by comparing the FPGA outputs with our Sage reference implementation (using the same PRNG and random seeds).

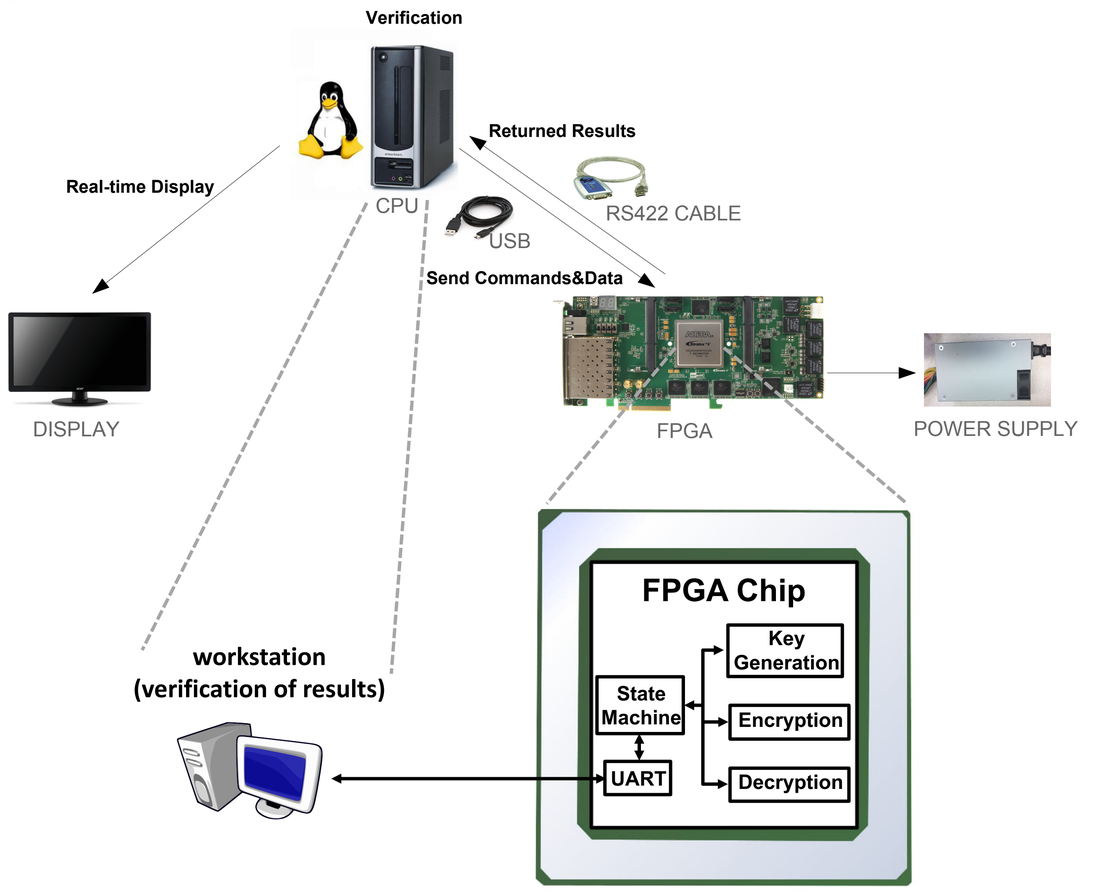

Fig.2: Evaluation setup

6. Performance

Case	Cycles		Logic	Mem.	Reg.	Fmax
	KeyGen.	Dec.				
area	11,121,214	34,492	53,447 (23%)	907 (35%)	118,243	245 MHz
bal.	3,062,936	22,768	70,478 (30%)	915 (36%)	146,648	251 MHz
time	966,400	17,055	121,806 (52%)	961 (38%)	223,232	248 MHz

Table 1: Performance for the entire Niederreiter cryptosystem (i.e., key generation, encryption, and decryption) including the serial IO interface when synthesized for the Stratix V (5SGXEA7N) FPGA

Design	Cycles			Logic	Freq. (MHz)	Mem.	Time (ms)		
	KeyGen.	Dec.	Enc.				Gen.	Dec.	Enc.
m = 11, t = 50, n = 2048, Virtex 5 LX110									
[SWM10]	14,670,000	210,300	81,500	14,537(84%)	163	75	90.00	1.29	0.50
This design	1,503,927	5,864	1,498	6,660(38%)	180	68	8.35	0.03	0.01
m = 12, t = 66, n = 3307, Virtex 6 LX240									
[MBR15]	—	28,887	—	3307	162	15	—	0.18	—
This design	—	10,228	—	6571	267	23	—	0.04	—
m = 13, t = 128, n = 8192, Hawell vs. Stratix V									
[Chou17]	1,236,054,840	343,344	289,152	—	4,000	—	309.0	0.09	0.07
This design	1,173,750	17,140	6,528	129,059(54%)	231	1,126	5.08	0.07	0.07

Table 2: Comparison with related work.

[SWM10] Shoufan, Abdulhadi, et al. "A novel cryptoprocessor architecture for the McEliece public-key cryptosystem." *IEEE Transactions on Computers* 59.11 (2010): 1533-1546.
[MBR15] Massolino, Pedro Maat C., Paulo SLM Barreto, and Wilson V. Ruggiero. "Optimized and scalable co-processor for McEliece with binary Goppa codes." *ACM Transactions on Embedded Computing Systems (TECS)* 14.3 (2015): 45.
[Chou17] Chou, Tung. "McBits revisited." *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, Cham, 2017.

7. Security and Design Parameters

Fully tunable design by use of code generation scripts.

All security parameters (m, t, n) can be freely chosen.

Performance parameters for controlling hardware parallelism:

- Compact, low-area design for embedded systems, …
- Large, high-performance design for server accelerator, …

8. Acknowledgements

This work was supported in part by United States' National Science Foundation grant 1716541.

9. Publications

- **Wen Wang**, Jakub Szefer, and Ruben Niederhagen, "FPGA-based Niederreiter Cryptosystem using Binary Goppa Codes" in *Proceedings of International Conference on Post-Quantum Cryptography (PQCrypto)*, April 2018.
- **Wen Wang**, Jakub Szefer, and Ruben Niederhagen, "FPGA-based Key Generator for the Niederreiter Cryptosystem using Binary Goppa Codes" in *Proceedings of the Conference on Cryptographic Hardware and Embedded Systems (CHES)*, September 2017.
- **Wen Wang**, Jakub Szefer, and Ruben Niederhagen, "Solving Large Systems of Linear Equations over GF(2) on FPGAs" in *Proceedings of the International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, November 2016.