1 Performance

This document reports the performance of additional implementations for BIKE1, BIKE2 and BIKE3. The performance is reported in processor cycles (lower is better), reflecting the performance per a *single core*. For each benchmark, the process was executed 25 times to warm-up the caches, followed by 100 iterations that were clocked (using the RDTSC instruction) and averaged. To minimize the effect of background tasks running on the system, each such experiment was repeated 10 times, and averaged. The results are reported in Tables 1, 2, and 3 for BIKE-1, Tables 4, 5, and 6 for BIKE-2, and in Tables 7, 8, and 9 for BIKE-3.

The implementation code. The core functionality was written in x86 assembly, and wrapped by assisting C code. The implementations use the PCLMULQDQ, AES−NI and the AVX2 and AVX512 architecture extensions. The code was compiled with gcc (version 5.4.0) in 64-bit mode, using the "O3" Optimization level, and run on a Linux (Ubuntu 16.04.3 LTS) OS. Details on the implementation and optimized components are provided in [1], and the underlying primitives are available in [2].

The benchmarking platform. The experiments were carried out on a platform equipped with the latest 8^{th} Generation Intel® Core^{TM} processor ("Kaby Lake") - Intel® Xeon® Platinum 8124M CPU at 3.00 GHz Core® i5 − 750. The platform has 70 GB RAM, 32K L1d and L1i cache, 1,024K L2 cache, and 25,344K L3 cache. It was configured to disable the Intel® Turbo Boost Technology, and the Enhanced Intel Speedstep® Technology.

References

[1] Nir Drucker and Shay Gueron. A toolbox for software optimization of qcmdpc code-based cryptosystems. Cryptology ePrint Archive, December 2017. http://eprint:iacr:org/.

[2] Shay Gueron. A-toolbox-for-software-optimization-of-qc-mdpc-code-basedcryptosystems, 2017. https://github:com/Shay-Gueron/A-

toolbox-for- software-optimization-of-QC-MDPC-code-based-cryptosystems.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	0.09	0.11	0.45	0.23	0.15	1.74
AVX512	0.09	0.11	0.40	0.23	0.13	1.57

Table 1: Performance (in millions of cycles) of BIKE1-64.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	0.25	0.28	1.01	0.54	0.36	4.59
AVX512	0.25	0.27	0.97	0.49	0.33	4.07

Table 2: Performance (in millions of cycles) of BIKE1-96.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	0.25	0.29	2.75	0.67	0.42	9.84
AVX512	0.25	0.27	2.24	0.69	0.36	8.27

Table 3: Performance (in millions of cycles) of BIKE1-128.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	4.38	0.09	0.41	4.51	0.12	1.80
AVX512	4.38	0.08	0.39	4.38	0.11	1.59

Table 4: Performance (in millions of cycles) of BIKE2-64.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	7.77	0.17	1.00	8.04	0.27	4.66
AVX512	7.79	0.17	0.76	8.05	0.23	4.07

Table 5: Performance (in millions of cycles) of BIKE2-96.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	11.99	0.27	2.70	12.45	0.39	10.74
AVX512	11.99	0.25	2.17	12.34	0.34	8.93

Table 6: Performance (in millions of cycles) of BIKE2-128.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	0.07	0.15	0.58	0.21	0.21	2.64
AVX512	0.07	0.14	0.43	0.20	0.19	2.44

Table 7: Performance (in millions of cycles) of BIKE3-64.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	0.16	0.32	1.47	0.42	0.45	7.41
AVX512	0.16	0.31	1.24	0.40	0.40	6.66

Table 8: Performance (in millions of cycles) of BIKE3-96.

	—			Constant time implementation		
	KeyGen	Encaps	Decaps	KeyGen	Encaps	Decaps
AVX2	0.25	0.50	3.05	0.81	0.78	13.62
AVX512	0.25	0.48	2.57	0.68	0.67	11.79

Table 9: Performance (in millions of cycles) of BIKE3-128.