Implementation and Evaluation of Toeplitz Strong Extractor for Post-Processing of QRNGs on Various Hardware Systems

ANURAG K S V* | RAGHAVAN G# | KANAKA RAJU P†

*Master's Student, #Director & Professor, †Assistant Professor at School of Quantum Technology, Defence Institute of Advanced Technology, Pune-411025, MH, India

INTRODUCTION

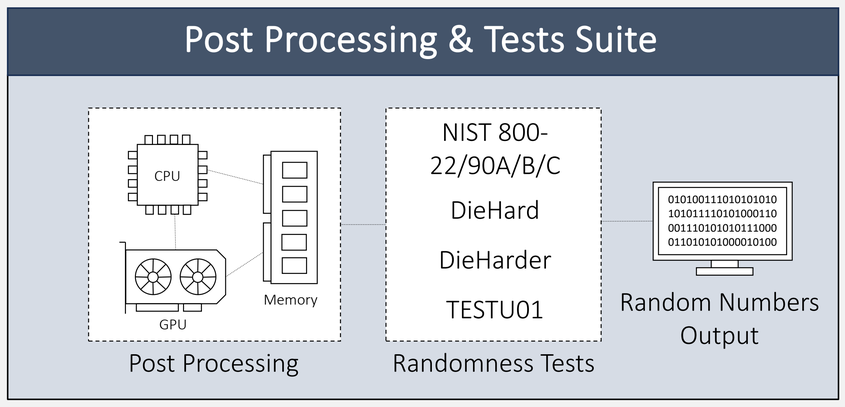
Post Processing of QRNG Raw Data

- To distil Quantum Randomness from Quantum + Classical Noise
- To generate information-theoretically provable Quantum Random Numbers

Post Processing Methods

- Matrix Multiplication based Toeplitz Hashing
- Fast Fourier Transform based Toeplitz Hashing

Ma X et. al., Physical Review A, 87(6), 062327, (2013)



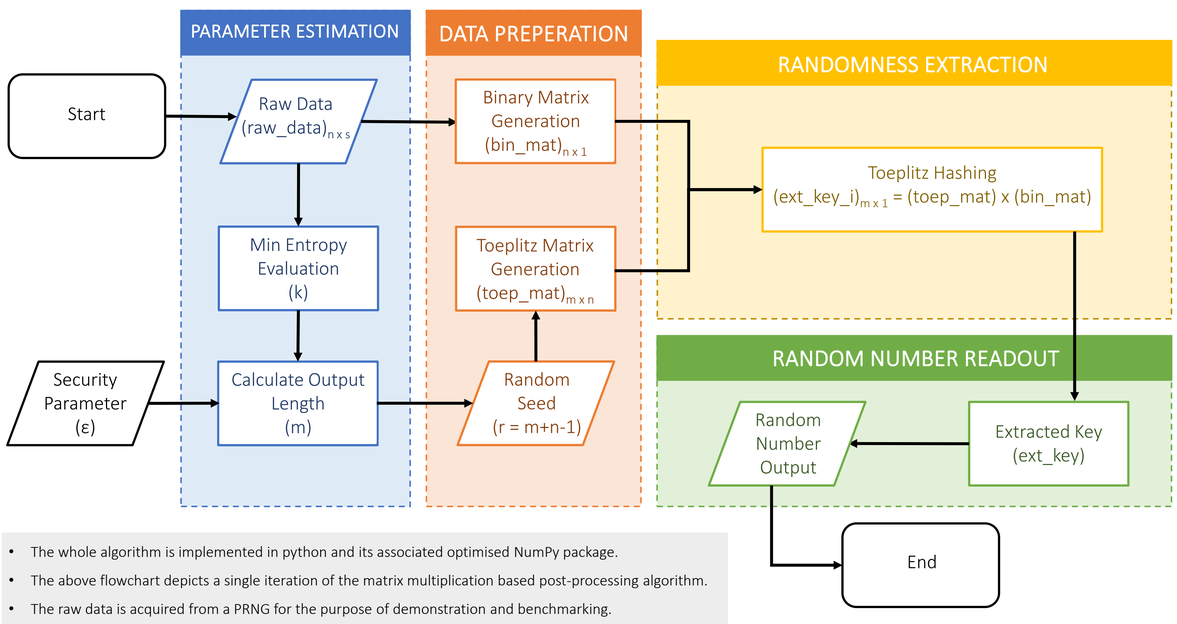
Hardware System Specifications		
Processor	AMD Ryzen 5 4600H 6C 12T	Intel Xeon Gold 6226R 16C 32T
Graphics Card	Nvidia GTX1650(M) 4 GB	Nvidia Quadro GV100 32 GB
RAM	DDR4 3200MHz 16GB	DDR3 1866 MHz 64 GB

Algorithmic Schematic for QRNG Post-Processing

Parameter Estimation → Data Preparation → Randomness Extraction → Random Number Readout

ALGORITHMIC FLOW

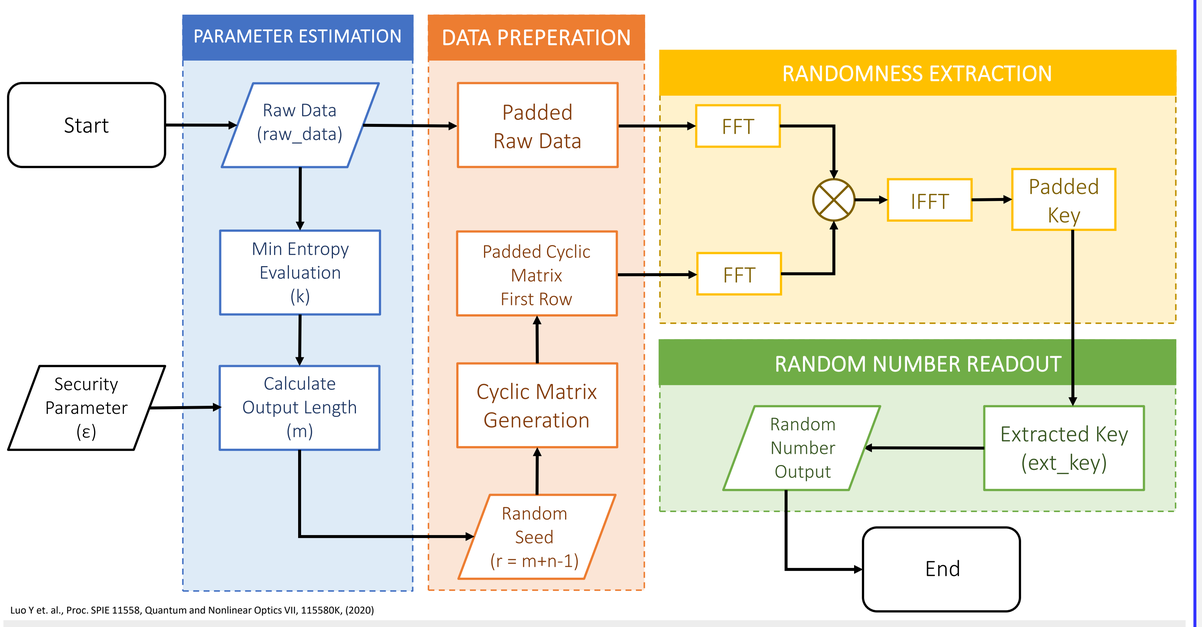
Matrix Multiplication based Toeplitz Hashing



- The whole algorithm is implemented in python and its associated optimised NumPy package.
- The above flowchart depicts a single iteration of the matrix multiplication based post-processing algorithm.
- The raw data is acquired from a PRNG for the purpose of demonstration and benchmarking.
- The output length is calculated in accordance to the Leftover Hash Lemma.

Zhang X et. al., IEEE-NPSS Real Time Conference (RT), Padua, Italy, pp. 1-5, (2016)

Fast Fourier Transform based Toeplitz Hashing

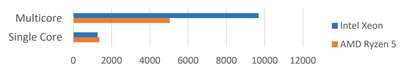


Luo Y et. al., Proc. SPIE 11558, Quantum and Nonlinear Optics VII, 115580K, (2020)

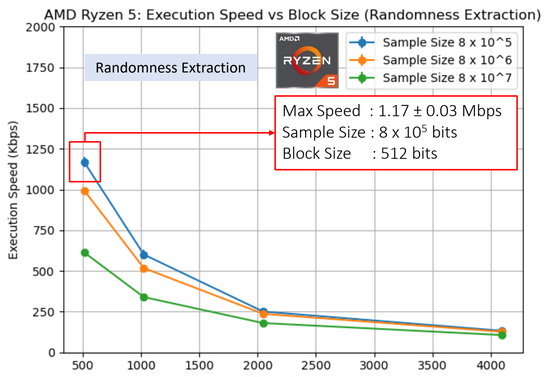
- The operation of matrix multiplication is converted to fast convolution based on the properties of the cyclic matrix.
- The GPU based FFT implementation used is highly optimized for 2^n inputs due to which we pad both the raw data and the first row of the cyclic matrix to the closest 2^n number.

BENCHMARKING AND RESULTS

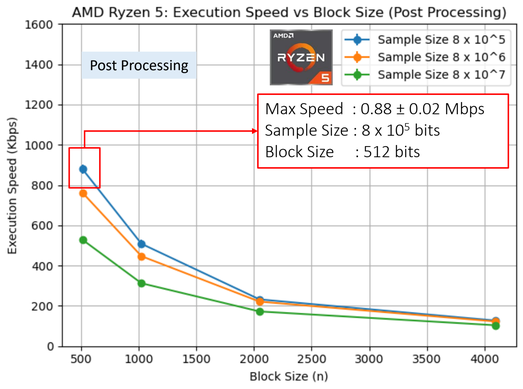
Matrix Multiplication based Toeplitz Hashing

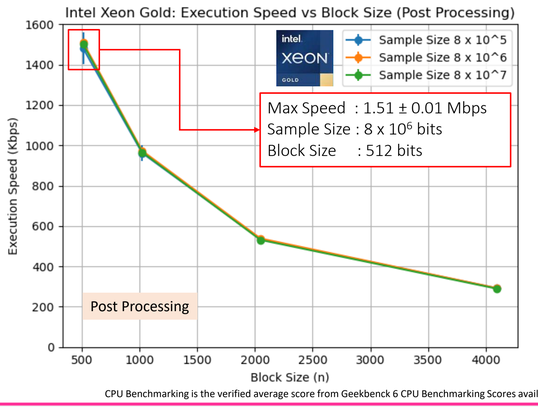


AMD Ryzen 5 based CPU



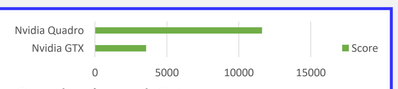
Intel Xeon Gold based CPU



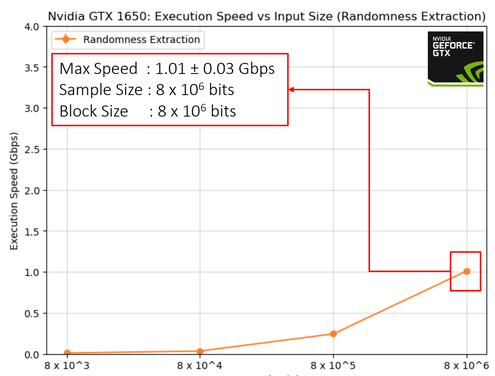


CPU Benchmarking is the verified average score from Geekbenck 6 CPU Benchmarking Scores available online.

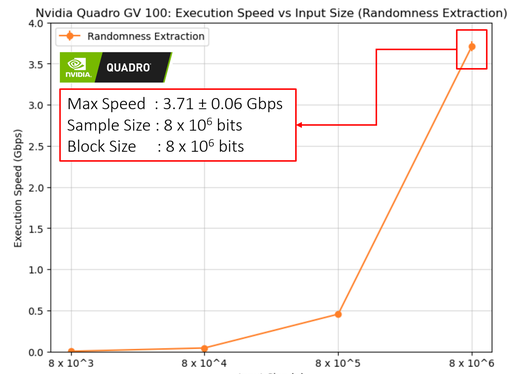
Fast Fourier Transform Based Toeplitz Hashing

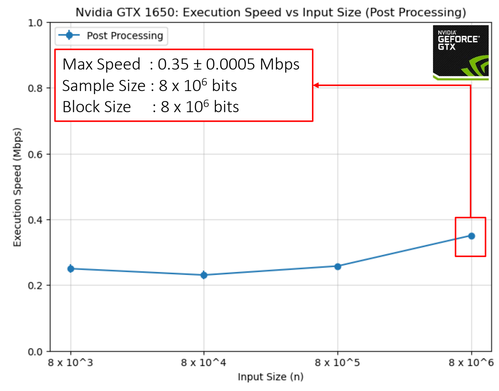


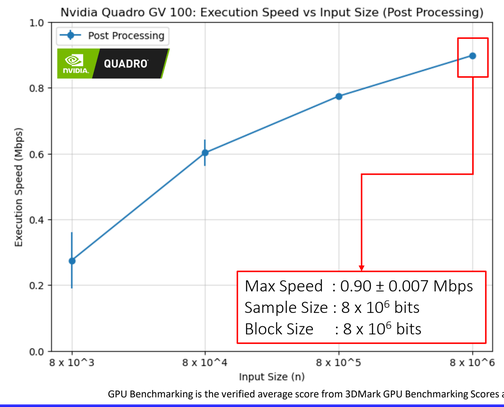
Nvidia GTX based GPU



Nvidia Quadro based GPU







GPU Benchmarking is the verified average score from 3DMark GPU Benchmarking Scores available online.

CONCLUSION

- We systematically elucidate the various implementation steps of the post-processing algorithm for the QRNG system.
- We have significantly improved the time and space complexity of the randomness extraction step by moving from matrix multiplication based Toeplitz Hashing to FFT based implementation of Toeplitz Hashing.
- We provide a detailed comparison of various implementation methods on different hardware systems, achieving a **maximum speed of 3.7 Gbps for Randomness Extraction** in a single iteration on Nvidia Quadro GV100 GPU.

Method Used	Time Complexity	Space Complexity	Hardware (CPU / GPU)	Sample Size (l x s)	Block Size (n)	Speed of Execution
Matrix Multiplication	$O(n^2)$	$O(n*m)$	AMD Ryzen 5 4600H	8×10^5	512	1.17 Mbps
			Intel Xeon Gold 6226R	8×10^7	512	1.89 Mbps
Fast Fourier Transform	$O(n.\log_2 n)$	$O(n+m-1)$	Nvidia GTX1650 (M)	8×10^6	8×10^6	1.01 Gbps
			Nvidia Quadro GV100	8×10^6	8×10^6	3.71 Gbps

email: anurag.krovvidi@gmail.com
raju@diat.ac.in

website: https://sqt.diat