CRYPTACUS Nov 2017

Quam Bene Non Quantum

Statistical analysis of IDQ's Quantis Quantum Random Number Generator

Darren Hurley-Smith & Julio Hernandez-Castro

d.hurley-smith@kent.ac.uk, jch27@kent.ac.uk

University of Kent

Introduction

- Darren Hurley-Smith
  - Research Associate
  - School of Computing, University of Kent
- Julio Hernandez-Castro
  - Professor
  - School of Computing, University of Kent

- Current research related to this presentation:
  - Analysing the properties of random number generators
  - Evaluating the effectiveness of certification schemes
  - Identifying independent and robust tests of randomness

Overview of this Research

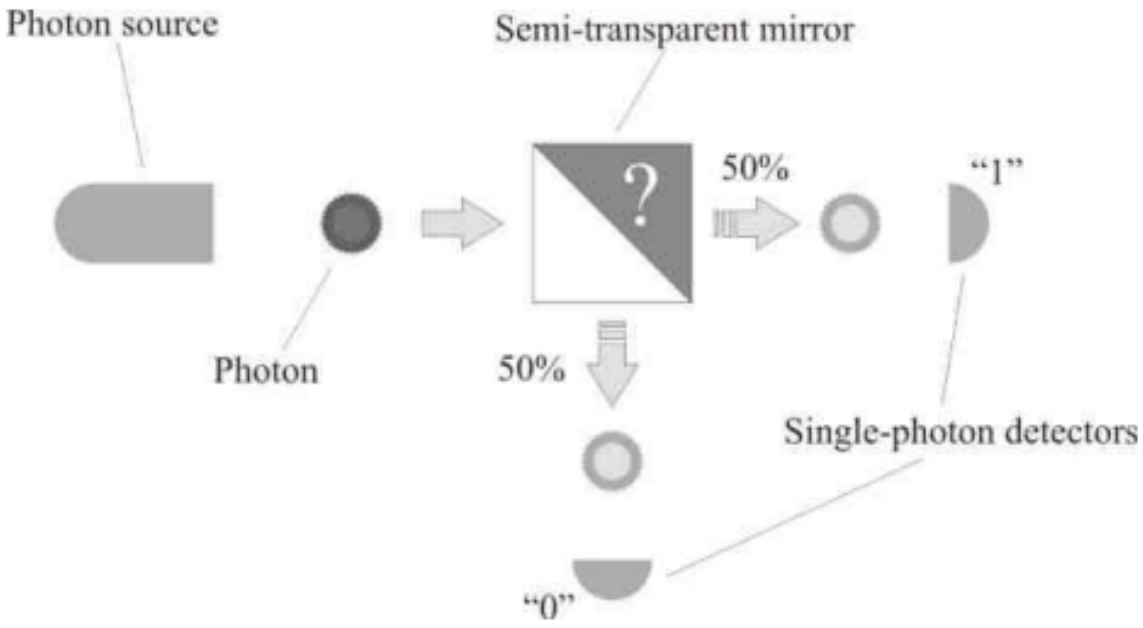
- Previously we've focused on TRNG
  - DESFire EV1 and EV2
  - ChaosKey, Araneus II, TRNG9815, and others…
- We've found flawed TRNG implementations
  - DESFire EV1
  - HotBits
- Optical Quantum randomness
  - Offers a tamper-proof phenomenon as a source of entropy
  - High-cost, high-speed and claims of high-quality
- What are the attributes of QRNG
  - Does it exhibit any entropy-source biases?
  - What best practices are involved in entropy collection?

IDQ Quantis QRNG Modules

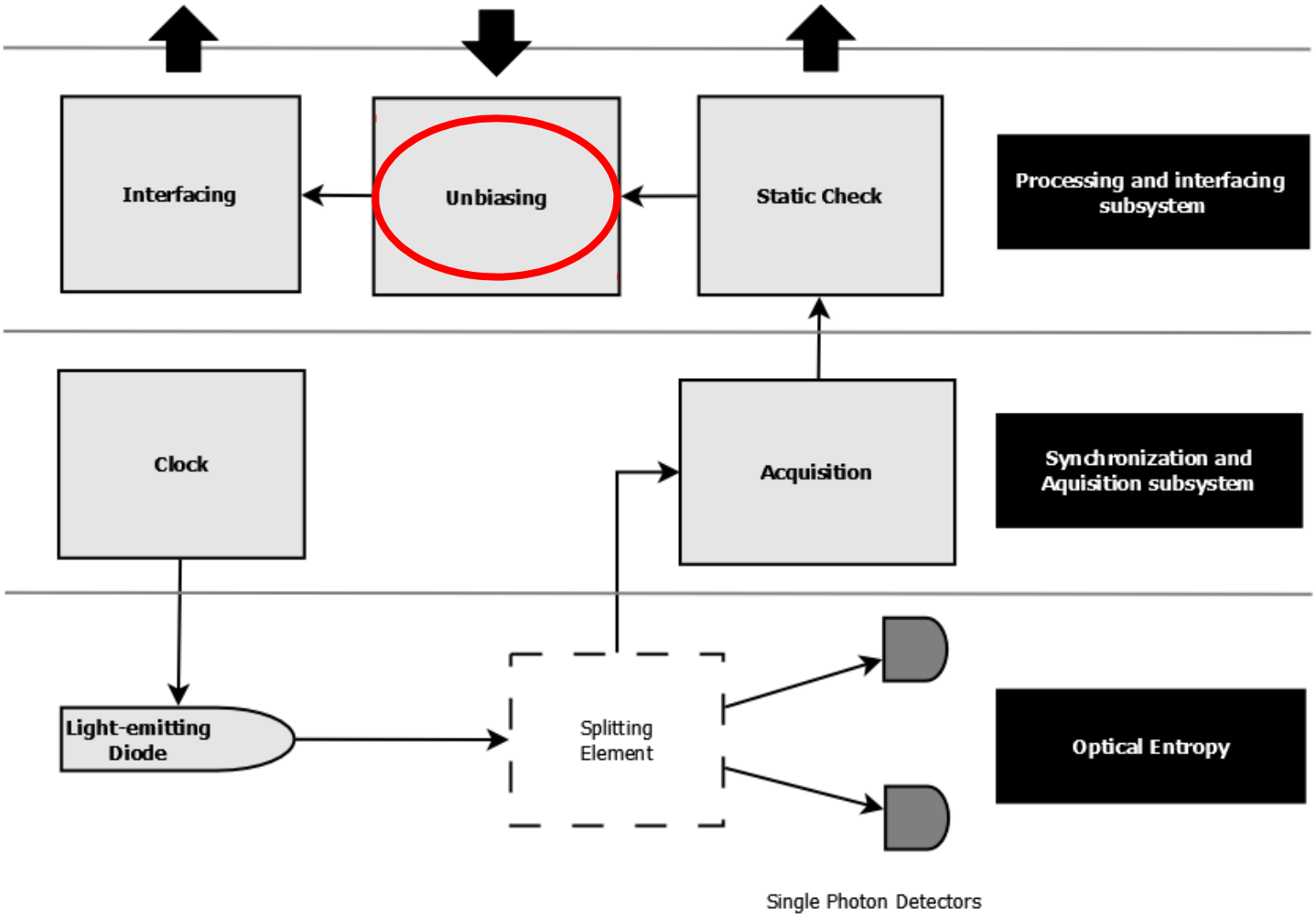
- Three modules tested:
  - 16M (PCI-E 16Mb/s output speed - top)
    - €2990
  - 4M (PCI-E 4Mb/s output speed - middle)
    - €1299
  - USB (4Mb/s output speed – bottom)
    - €990
- Certification provided:
  - Self-certification (Dieharder, NIST SP800-22)
  - Compliance Testing Lab certified (UK)
  - METAS certified
- Real world use-case:
  - Swiss Loterie Romande

Optical Quantum Random Number Generation

- The entropy source pictured below is common to all Quantis devices



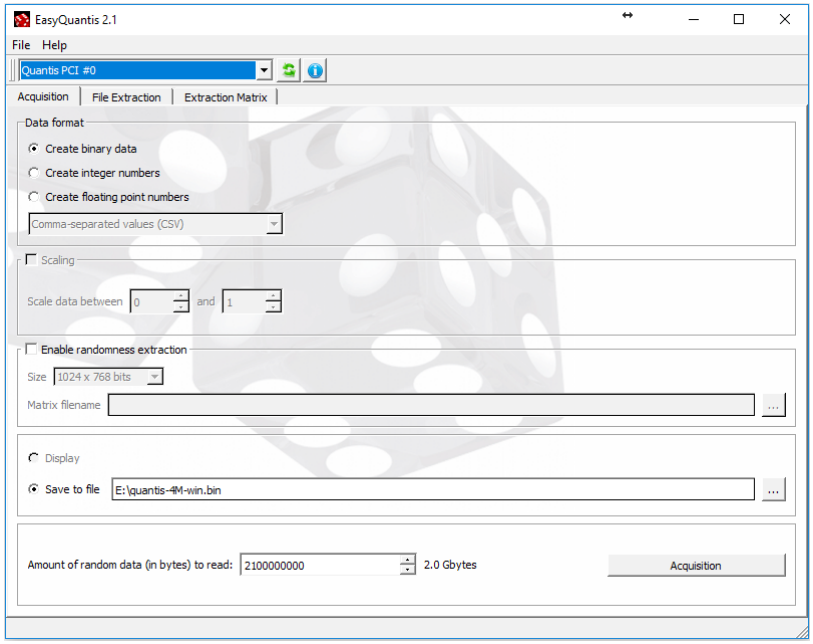
Quantis QRNG Block Diagram



Experiment Set-up

- Initial Data-collection:
  - 3 x 2GiB files collected from each device
  - EasyQuantis GUI Application used to collect data
  - No post-processing

- A more thorough follow-up:
  - 100 x 2GiB collected from each device
  - EasyQuantis command line utility used for collection
  - Raw and post-processed data



Experiment Set-up 2

Results

- Speedtest
  - 16M (15.87Mb/s), 4M (3.86Mb/s), USB (3.96Mb/s)
  - ChaosKey TRNG (3.8Mb/s)
- Dieharder
  - Initial tests show some issues
  - Larger sets show this to be a statistical error
- NIST SP800-22 (STS2.1.2)
  - Initial tests show some issues
  - Larger sets show this to be a statistical error
- ENT
  - Significant byte-level biases found
  - Serial correlation of bits shows some poor results
- TestU01
  - Alphabits reports multiple failures
  - Rabbit battery reports multiple failures

Detailed Results (Dieharder/NIST/TestU01)

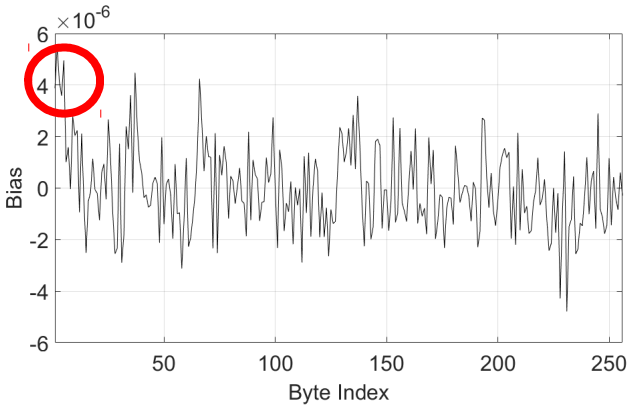
Device	Size	Dieharder	NIST STS 2.1.2	Alphabits	Rabbit
	(GiB)	(Failed/Weak/Passed)	(Passed/Total)	(Passed/Total)	(Passed/Total)
Quantis 16M	2	8 / 11 / 95	182 / 186	7 / 17	26 / 40
	2	6 / 13 / 95	181 / 186	9 / 17	32 / 40
	2	7 / 11 / 96	182 / 186	7 / 17	29 / 40
Quantis 4M	2	0 / 3 / 111	185 / 186	7 / 17	28 / 40
	2	0 / 5 / 109	186 / 186	7 / 17	28 / 40
	2	0 / 6 / 108	186 / 186	7 / 17	27 / 40
	16	0 / 4 / 110	N/A	5 / 17	25 / 40
	32	0 / 3 / 111	N/A	5 / 17	25 / 40
	200	0 / 2 / 112	N/A	5 / 17	24 / 40
Quantis USB	2	0 / 6 / 108	184 / 186	14 / 17	33 / 40
	2	0 / 7 / 107	186 / 186	11 / 17	29 / 40
	2	1 / 6 / 107	184 / 186	10 / 17	30 / 40
ChaosKey	2	0 / 3 / 111	184 / 186	17 / 17	40 / 40
/dev/urandom	2	0 / 3 / 111	186 / 186	17 / 17	40 / 40

- Dieharder and NIST are passed
  - 16M is an exception, but further testing suggests these three initial results are anomalous
- Alphabits and Rabbit fail consistently
  - Devices fail slightly different tests more frequently than others
  - ChaosKey (TRNG USB module) passes all tests providing a TRNG baseline
  - */dev/urandom* also passes all tests providing a PRNG baseline

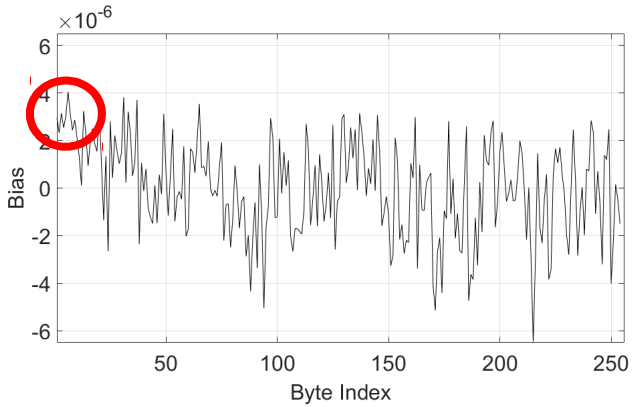
TestU01 Results in Detail

- Alphabits
  - 4M fails: MultinomialBitsOver, HammingIndep, RandomWalk
  - 16M fails: MultinomialBitsOver, RansomWalk
  - USB fails: MultinomialBitsOver, HammingIndep, RandomWalk
- Rabbit
  - 4M fails: HammingWeight, AutoCor, Run of Bits, RandomWalk
    - 4M fails almost every permutation of the RandomWalk test
  - 16M fails: HammingWeight, AutoCor, Run of Bits, RandomWalk
    - 16M fails fewer permutations of each test
  - USB fails: Fourier3, HammingWeight, HammingIndep, AutoCor, Run of Bits, RandomWalk
    - Unlike Alphabits, the Rabbit battery shows some differences between USB and 4M

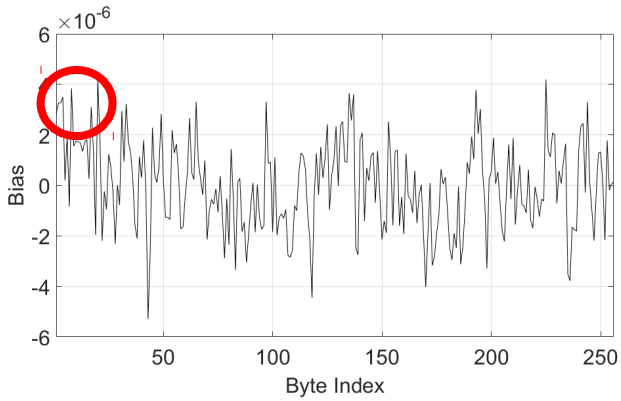
ENT Results in Detail



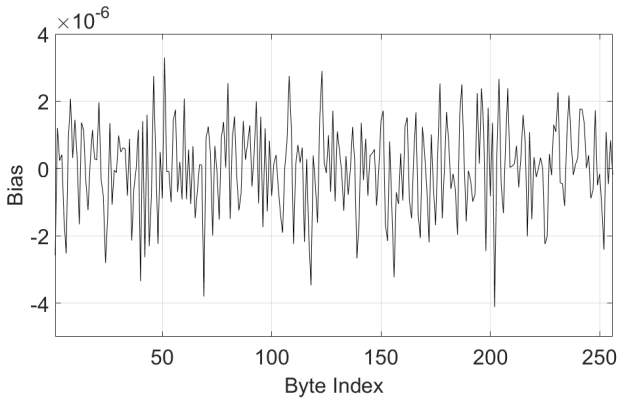
Quantis 16M Bias



Quantis 4M Bias



Quantis USB Bias



/dev/urandom Bias

ENT Results in Detail 2

File	Size (Bytes)	X^2
16M 1	2100000000	373.1276
16M 2	2100000000	302.5997
16M 3	2100000000	354.927
16M 4	2100000000	344.8863
16M 5	2100000000	350.1756
16M 6	2100000000	300.5472
16M 7	2100000000	333.1116
16M 8	2100000000	349.3203
16M 9	2100000000	307.7769
16M 10	2100000000	323.3886

16M byte-level Ent

16M bit-level Ent

File	Size (Bits)	X^2
16M 1	1.68E+10	107.7214
16M 2	1.68E+10	71.43077
16M 3	1.68E+10	99.39572
16M 4	1.68E+10	71.63044
16M 5	1.68E+10	67.33272
16M 6	1.68E+10	89.23194
16M 7	1.68E+10	60.23114
16M 8	1.68E+10	68.17352
16M 9	1.68E+10	47.17763
16M 10	1.68E+10	62.60946

ENT Results in Detail 3

File	Size (Bytes)	X^2	Serial Correlation	HTCC	p-value
4M 1	2097152000	553.7709	1.40886E-05	0.645181505	0.74030
4M 2	2097152000	489.7084	-3.87639E-05	1.775179538	0.96146
4M 3	2097152000	514.2832	-1.63582E-05	0.749119473	0.77276
4M 4	2097152000	510.2955	5.11633E-05	2.343007275	0.99005
4M 5	2097152000	487.6929	-2.57854E-05	1.180832882	0.88061
4M 6	2097152000	506.4829	1.62732E-05	0.745227468	0.77158
4M 7	2097152000	440.1236	4.11375E-05	1.883879676	0.96964
4M 8	2097152000	548.0064	3.53198E-05	1.617459493	0.94649
4M 9	2097152000	460.4062	-1.63436E-05	0.748449739	0.77256
4M 10	2097152000	540.5753	3.50277E-05	1.60408015	0.94503

4M byte-level Ent

4M bit-level Ent

File	Size (Bits)	X^2	Serial Correlation	HTCC	p-value
4M 1	16777216000	141.8065	9.36056E-05	12.13267346	0.97382
4M 2	16777216000	103.5581	8.43226E-05	10.9294641	0.97096
4M 3	16777216000	140.0126	8.73252E-05	11.31864453	0.97195
4M 4	16777216000	78.39836	9.30651E-05	12.0544358	0.97365
4M 5	16777216000	38.90776	0.000100731	13.04736851	0.97565
4M 6	16777216000	36.76245	0.000106742	13.82597243	0.97702
4M 7	16777216000	40.74031	9.87896E-05	12.79591623	0.97517
4M 8	16777216000	81.78343	9.64242E-05	12.48953167	0.97457
4M 9	16777216000	44.9761	9.35218E-05	12.11358506	0.97378
4M 10	16777216000	53.16695	9.94391E-05	12.88003447	0.97534

ENT Results in Detail 4

File	Size (Bytes)	X^2	Serial Correlation	HTCC	p-value
USB 1	2097152000	436.9514	3.19571E-05	1.464460426	0.92785
USB 2	2097152000	414.8936	1.58698E-05	0.72724485	0.76612
USB 3	2097152000	480.2678	-5.93089E-06	0.271787582	0.60699
USB 4	2097152000	476.2802	-4.53518E-06	0.207828094	0.58223
USB 5	2097152000	431.4787	-4.63862E-05	2.125682795	0.98275
USB 6	2097152000	489.5561	-1.59544E-05	0.731120809	0.76731
USB 7	2097152000	519.5624	1.54902E-05	0.709851536	0.76077
USB 8	2097152000	455.8573	3.17628E-05	1.455555416	0.92662
USB 9	2097152000	440.9796	-1.37956E-05	0.632191614	0.73608
USB 10	2097152000	443.5124	6.59489E-06	0.302215889	0.61863

USB byte-level Ent

USB bit-level Ent

File	Size (Bits)	X^2	Serial Correlation	HTCC	p-value
USB 1	16777216000	46.03617	8.88723E-05	11.51916149	0.97243
USB 2	16777216000	20.5772	8.75407E-05	11.346569	0.97201
USB 3	16777216000	48.20293	9.39057E-05	12.17157021	0.97390
USB 4	16777216000	37.68492	9.72978E-05	12.61123074	0.97481
USB 5	16777216000	42.02821	8.59015E-05	11.13411296	0.97148
USB 6	16777216000	40.94392	9.69857E-05	12.57077812	0.97473
USB 7	16777216000	49.70702	9.89094E-05	12.82012646	0.97522
USB 8	16777216000	57.68654	8.80525E-05	11.41291091	0.97218
USB 9	16777216000	48.25115	8.28145E-05	10.73398723	0.97043
USB 10	16777216000	20.40718	9.91219E-05	12.84766418	0.97527

Discussion

- Quantis raw output is obviously biased
  - IDQ respond stating that post-processing is required
  - Post-processing is listed as optional in their product documentation
  - Robust randomness at 16Mb/s is claimed:
    - Post-processing cuts this by 70%
- Post-processing solves all identified problems
  - Software implementation means that trust must be placed in both device and independent software
- Suitability for IoT is in question
  - Post-processing is CPU and memory intensive
- Yet more evidence of lax certification
  - Diehard over 10x1MB of data garnered official approval
  - Dieharder and NIST indicate serious problems, but miss flaws
  - Better, independent tests must be identified

Key Takeaways

- Quantum randomness is over-hyped
  - Products are costly, promise too much
  - Better NQ products at a fraction of the price
- IoT QRNGs will likely not be a reality anytime soon
  - Skepticism over opposing claims advised
- Post-processing in QRNGs is never optional
  - And should be implemented in hardware
- Self certification is meaningless and suspicious
  - Certification is not much better, but more rigorous

Thank you for listening!

Questions?

github.com/DHSatUNIKENT/CRYPTACUS_Quantis_Results

THE UK'S EUROPEAN UNIVERSITY

www.kent.ac.uk

University of Kent