Quantum Security for Cloud and Virtualised Environments

David Wilson

Farid Boussaid[1], Mohammed Bennamoun[2]
1. Electrical, Electronic and Computer engineering,
2. Computer Science and Software Engineering
The University of Western Australia

Shelton Wilson
CEED Client: Research Tech Pty Ltd

Abstract

Quantum random number generators (QRNGs) are novel entropy sources that provide information theoretic security for encryption key generation and seeding. Entropy, in this context, refers to the measure of unpredictability or randomness; higher entropy means greater unpredictability, which is crucial for robust encryption. Insufficient entropy can compromise encryption, a problem more widespread than expected. A particular concern is the industry's rapid transition from on-premises to cloud and virtualized deployments, where entropy scarcity is compounded. Hence, there are significant benefits in adapting QRNG technology for the cloud. This project investigates certain aspects of this goal including the testing of QRNGs, thermal management, using buffering to improve performance, and implementing continuous health tests for continuous reliability monitoring.

1. Introduction

Quantum random number generators (QRNGs) offer a novel solution to the problem of 'entropy scarcity', which can undermine the security of seeding and key generation reducing the effectiveness of encryption in computer systems. Entropy is a measure of disorder or 'randomness'; computer systems accumulate entropy to ensure the strength of encryption. However, secure sources of entropy are increasingly scarce, especially on cloud or virtualised platforms. The scarcity of entropy has been shown to lead to the generation of weak keys. While the guarantee of computational security offered by cryptographically secure pseudorandom number generators (CSPRNGs) should be sufficient, this security is only attained if the generator is seeded with sufficient entropy. A concerning example is the unexpected prevalence of common factors in RSA (Rivest-Shamir-Alderman) public keys (Heninger et al., 2012; Lenstra et al., 2012), with as many as 0.5% of surveyed TLS (transport layer security) public moduli offering no security due to common factors (Heninger et al., 2012). This is striking since there are about 10^{305} RSA-1024 primes (Goldstein, 1973), meaning common factors should never occur if primes are selected uniformly. The same work found that duplicate ephemeral DSA (digital signature algorithm) keys were also common (representing 1.6% of the sample size). These failures were attributed to a lack of entropy, particularly during boot time, when keys are often generated (Heninger et al., 2012). The security of encryption depends entirely on the unpredictability of the secret key (Kerckhoffs,

1883), however, this is inherently challenging since computers are deterministic and cannot produce true random numbers (Von Neumann, 1963). Operating systems accumulate entropy from weak entropy sources such as interrupt timing (Atsec information security GmbH, 2020). However, it is impossible to reliably estimate how much entropy these sources contribute (Corrigan-Gibbs & Jana, 2015). Furthermore, the security of such sources assume that the attacker has no way of observing or influencing interrupt timings which is not the case. In fact, this is an instance of security by obscurity which is not accepted as a sound practice (Shannon, 1949). Systems are particularly insecure shortly after boot as keys may be generated before sufficient entropy has been collected. These issues are even more important for cloud and virtualised environments, which are low entropy environments (Fernandes et al., 2013) and are vulnerable to additional attacks such as replay attacks (Everspaugh et al., 2014).

QRNGs are state-of-the-art entropy sources which utilise the irreducibility of quantum mechanical uncertainty to provide information-theoretically secure randomness. True randomness is non-deterministic, and therefore must arise from physical not algorithmic origins (Von Neumann, 1963). However, even physical systems can be analysed and predicted. Classically, all systems are deterministic, only apparent randomness arises from the chaotic nature of these systems. This does not apply to quantum systems, according to Heisenberg's uncertainty principle which argues that randomness is an inherent property of nature (Busch et al., 2007; Heisenberg, 1925; Heisenberg, 1927). The principle behind QRNGs arises from Born's rule which explains that observations of a quantum state are inherently random, meaning that measurements of quantum states are a source of true randomness (Ma et al., 2016). While the outcome of a classical process can be modelled and predicted by an attacker with sufficient computational resources and knowledge, quantum mechanical uncertainty cannot be reduced and the resulting random sequence is formally incomputable (Calude et al., 2010), hence QRNGs can be said to provide information theoretic security (Shannon, 1949).
Many QRNG devices are commercially available (Herrero-Collantes & Garcia-Escartin, 2017), but a barrier to widespread adoption is the problem of entropy distribution – a method is required to distribute entropy from a QRNG device to physical/virtual servers. Entropy as a Service (EaaS) is one model being explored where entropy from a QRNG is distributed to multiple clients over a network (Vassilev & Staples, 2016). However, this exposes the QRNG seeding process to network-based attack. This project is being conducted in collaboration with Research Tech Pty Ltd which is exploring an entropy distribution method that would allow multiple tenanting virtual machines (VMs) to communicate to a single QRNG device – current QRNG cards will only allow a single VM to be mapped to a virtual machine. This allows for entropy to be distributed to many virtual machines or cloud instances simultaneously.

1.1 Project objectives

The project aims to contribute to the development of QRNG technology. The objectives of this project include:

- To develop an IP core which implements the NIST (National Institute of Standards and Technology) continuous health tests for entropy sources which can be deployed to the FPGA (field gate programmable array) device within the QRNG.
- To develop an IP core which implements a mechanism which buffers the random data to DRAM (dynamic random access memory) to improve QRNG latency.
- To investigate the thermal management of the FPGA/QRNG device.
- To statistically validate the randomness of the generated output.

2. Methodology

2.1 IP core design

The IP cores are being designed using AMD Vivado for the verification, synthesis, and implementation of HDL (hardware description language) designs. The company requires the use of Vivado IP integrator (IPI) (AMD Xilinx, 2022) enabling facile re-use of HDL IP cores and streamlined handover of project deliverables. The IP cores utilise AXI4, AXI4-lite and AXI4-Stream protocols to ensure interoperability with the rest of the system. AXI4-Stream is a facile interface for the transfer of sequential unaddressed data. Transfers occur on the rising clock edge if and only if the slave asserts TREADY and the master asserts TVALID (ARM, 2010). AXI4 is memory mapped protocol allowing a master read/write access to a slave's address space. AXI4 is very high-throughput and requires a substantial amount of logic to implement. AXI4-lite is a subset of AXI4 suitable for area-efficient implementations which do not require high data rates (ARM, 2023).

2.2 NIST continous health tests

QRNGs are generally utilized when a greater assurance of secure key generation and seeding is required. In these situations, ensuring the correct operation of the QRNG device is critical. NIST special publication SP800-90B (Turan et al., 2018), which provides standardized test procedures for verifying the correct operation of entropy sources, specifies two kinds of health testing: on-demand tests and health tests. On-demand tests are in-depth tests designed to be applied once, usually upon start up or otherwise on-demand, to verify the correct operation of the entropy source. Continuous health tests are continuous online checks which are designed to detect the failure of entropy sources during operation.

In accordance with SP800-90B, two light-weight health tests are required, the repetition count tests and the adaptive proportions tests. The repetition count tests monitor the lengths of runs identical values from the entropy source. This test can quickly detect failure which results in the output becoming stuck at the same values or the total failure of the entropy source.

Since the min-entropy H_∞, for a distribution of i symbols, is given by (Kim et al., 2020),

$$H_\infty = -\log(\max p_i) \tag{1}$$

Then the probability p of any symbol occurring is bounded by,

$$p \leq 2^{-H_\infty} \tag{2}$$

Hence, the probability p^{n-1} of a run of n identical values is at most,

$$p^{n-1} \leq 2^{-H_\infty(n-1)} \tag{3}$$

The test maintains a counter to track the length of consecutive runs, if the count c exceeds a threshold n then the test fails. It is possible to choose threshold n based on H_∞ and an acceptable false positive rate α as follows (Turan et al., 2018),

$$n = 1 + \left\lceil \frac{\log_2 \alpha}{H_\infty} \right\rceil \tag{4}$$

The adaptive proportions tests (Turan et al., 2018), unlike the repetition count test which detects complete failures of the entropy source, is designed to detect faults which result in degraded operation. The test is conducted over a sampling window, W during which the number of reoccurrences of the 1st sampled value over the remaining $(W - 1)$ samples is recorded. If the count exceeds a threshold, the test fails. As before, the probability of

reoccurrence is given by (2). Therefore, the distribution of sample may be approximate by a binomial distribution,

$$X \sim Bin(W - 1, 2^{-H\infty}) \quad (5)$$

Thus, it is trivial to select a threshold that will have a suitable false positive rate using the above distribution.

2.3 Statistical validation of randomness

To verify the correct operation of the QRNG it is necessary to statistically validate the QRNG output with the aim of identifying potential deviations from a random output. Statistical tests cannot demonstrate that a system is non-deterministic, however, they may detect correlations and biases which should not be present in QRNGs. Testing will employ both commonly reported methods such as the autocorrelation plot (Moeini et al., 2024; Park et al., 2019; Sanguinetti et al., 2014), and well-accepted test suites for random number generator such as NIST SP 800-22 (Rukhin et al., 2010) and Dieharder (Brown, 2006; Canonical, 2024).

3. Results and Discussion

3.1 IP core development

The NIST continuous health checks IP is being implemented in SystemVerilog and is capable of monitoring multiple entropy sources simultaneously. The core receives data via an AXI4-Stream interface and reports test failures over an AXI4 memory mapped lite interface. The DDR buffering IP core is currently also being developed. The core implements a FIFO-like (first-in, first-out) mechanism which can be backed by an AXI memory mapped compatible memory devices such as off-chip DRAM or on-chip memories.

3.2 Statistical validation of randomness

Statistical testing of the QRNG output has only just commenced, however, early results have not identified any statistical anomalies or defects in the output. The QRNG output was subjected to the Dieharder test suite which encompasses the NIST statistical test suite SP800-22, the Diehard tests (by G. Margsalia) and several original tests (Brown, 2006; Canonical, 2024) comprising a total of 174 tests. Each test returns either a pass or failure and weak failure results corresponding to 10^{-6} and 0.005 levels of significance respectively. The QRNG output returns a pass on all 174 tests. For comparison, the same test was run on /dev/urandom (the Linux RNG) and returned two weak failures. This is an early indication that the QRNG output distribution is uniform and unbiased.

4. Conclusions and Future Work

Quantum random number generators are a novel class of entropy sources which provide information theoretic security for key generation and seeding. Given the lack of entropy in cloud and virtualized environments, QRNG devices adapted for the cloud have potential to resolve security vulnerabilities affecting essentially all users of cloud-based services. Progress towards this aim is being made in terms of developing and verifying the NIST continuous health tests IP core and the DRAM buffering IP core. Initial test results of statistical validation are promising. However, more testing is required for the results to be conclusive. Performance of a heatsink solution for the QRNG device is currently being investigated. Future work may

comprise experimentally validating the simulation results to determine if this conclusion is correct.

5. Acknowledgements

I would like to thank the Department of Jobs, Tourism, Science and Innovation and Research Tech Pty Ltd for funding and support. Thanks, also to my supervisors, Prof. Farid Boussaid and Prof. Mohammed Bennamoun, for their guidance over the course of the project. I would like to thank Dr. Jeremy Leggoe and Kimberlie Hancock for their support in establishing this CEED project.

6. References

AMD Xilinx. (2022). *Vivado Design Suite User Guide: Designing IP Subsystems Using IP Integrator* (UG994).

ARM. (2010). *AMBA 4 AXI4-Stream Protocol Specification* (IHI 0051A ID030610).

ARM. (2023). *AMBA AXI Protocol Specification* (ARM IHI 0022).

Atsec information security GmbH. (2020). *Documentation and Analysis of the Linux Random Number Generator*.

Brown, R. G. (2006). *DieHarder: A Gnu Public License Random Number Tester*.

Busch, P., Heinonen, T., & Lahti, P. (2007). Heisenberg's uncertainty principle. *Phys. Rep.*, *452*(6), 155-176.

Calude, C. S., Dinneen, M. J., Dumitrescu, M., & Svozil, K. (2010). Experimental evidence of quantum randomness incomputability. *Physical Review A*, *82*(2), 22102.

Canonical. (2024). *Ubuntu Manpage: dieharder - A testing and benchmarking tool for random number generators*. Retrieved April from https://manpages.ubuntu.com/manpages/jammy/en/man1/dieharder.1.html

Corrigan-Gibbs, H., & Jana, S. S. (2015). Recommendations for Randomness in the Operating System, or How to Keep Evil Children out of Your Pool and Other Random Facts. USENIX Workshop on Hot Topics in Operating Systems,

Everspaugh, A., Zhai, Y., Jellinek, R., Ristenpart, T., & Swift, M. (2014). Not-So-Random Numbers in Virtualized Linux and the Whirlwind RNG. IEEE Symposium on Security and Privacy,

Fernandes, D. A. B., Soares, L. F. B., Freire, M. M., & Inácio, P. R. M. (2013). Randomness in Virtual Machines. Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing,

Goldstein, L. J. (1973). A History of the Prime Number Theorem. *Am. Math. Mon.*, *80*(6), 599-615.

Heisenberg, W. (1925). A quantum-theoretical reinterpretation of kinematic and mechanical relations. *Eur. Phys. J. A.*, *33*, 879-893.

Heisenberg, W. (1927). The actual content of quantum theoretical kinematics and mechanics. *Zeitschrift fur Physik*, *43*(3), 197-198.

Heninger, N., Durumeric, Z., Wustrow, E., & Halderman, J. A. (2012). Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. USENIX Security Symposium,

Herrero-Collantes, M., & Garcia-Escartin, J. C. (2017). Quantum random number generators. *Reviews of Modern Physics*, *89*(1), 48.

Kerckhoffs, A. (1883). La Cryptographie Militaire. *Journal des Sciences Militaires*, *9*(5).

Kim, Y., Guyot, C., & Kim, Y.-S. (2020). On the Efficient Estimation of Min-Entropy. *IACR Cryptol. ePrint Arch.*

Lenstra, A. K., Hughes, J. P., Augier, M., Bos, J. W., Kleinjung, T., & Wachter, C. (2012). Ron was wrong, Whit is right. *IACR Cryptol. ePrint Arch.*, 64.

Ma, X., Yuan, X., Cao, Z., Qi, B., & Zhang, Z. (2016). Quantum random number generation. *NPJ Quantum Inf.*, *2*(1), 16021.

Moeini, M., Akbari, M., Mirsadeghi, M., Naeij, H. R., Haghkish, N., Hayeri, A., & Malekian, M. (2024). Quantum random number generator based on LED. *J. Appl. Phys.*, *135*(8).

Park, B. K., Park, H., Kim, Y. S., Kang, J. S., Yeom, Y., Ye, C., Moon, S., & Han, S. W. (2019). Practical True Random Number Generator Using CMOS Image Sensor Dark Noise. *IEEE Access*, *7*, 91407-91413.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., & Vo, S. (2010). *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications* (SP 800-22 Rev 1a).

Sanguinetti, B., Martin, A., Zbinden, H., & Gisin, N. (2014). Quantum Random Number Generation on a Mobile Phone. *Phys. Rev. X*, *4*(3), 31056.

Shannon, C. E. (1949). Communication Theory of Secrecy Systems. *Bell System Technical Journal*, *28*(4), 656-715.

Turan, M. S., Barker, E., Kelsey, J., McKay, K. A., Baish, M. L., & Boyle, M. (2018). *Recommendation for the Entropy Sources Used for Random Bit Generation* (SP 800-90B).

Vassilev, A., & Staples, R. (2016). Entropy-as-a-Service: Unlocking the Full Potential of Cryptography. *Computer*, *49*(9), 98-102.

Von Neumann, J. (1963). Various techniques used in connection with random digits. *John von Neumann, Collected Works*, *5*, 768-770.