Supply Chain Attacks

Bjørn Mathias Helseth
Royal Holloway, University of London

Table of Contents

Supply Chain Attacks

Currently, Random Number Generator (RNG) manufacturers continuously use the statistical tests of randomness found in for example standards such as FIPS 140-2 to prove that their RNG is cryptographically secure. Even though this particular standard for statistical tests is now starting to become deprecated [1], it is still widely used in the supply chain by manufacturers to prove that their RNGs are producing output which can be used in cryptographical applications to provide a secure layer of randomness. How can we know that the claims of the manufacturer stand true at the point it is handed to the end-customer?

If we assume three phases of the process of distributing Hardware RNGs (HRNGs), just to provide an example of the level of trust signed by the end-customer to the RNG manufacturers. The manufacturing phase, the supply chain and then leaving the supply chain to the end-user or organisation. In these phases, we can assume that there are different vulnerabilities that can be introduced to the RNG in circulation.

We should be worried about these assumed vulnerabilities being present in the supply chain, so that we can prevent adversarial tampering of the RNGs. If the statistical tests are to be trusted, they should be able to detect adversarial bias. As Hurley-Smith et al. [2] states in their journal on the lightness of FIPS 140-2, “Our research shows that FIPS 140-2 cannot identify adversarial biases effectively, even very primitive ones.”. This can lead us to think that this commonly used statistical testing suite, can’t be trusted for use in testing the integrity of an RNG after it has been manufactured.

For statistical testing of randomness, there are generally a series of tests being performed on bit-blocks from the output with constant size. These tests look for particular distinctive characteristics usually found in random sequences. In the NIST SP800-22 and FIPS 140-2 batteries of tests for example, we find similar [2] [3] tests being used for general testing, such as the monobit test where we test to see if the number of ones and zeroes in a bit-block sequence are approximately the same as in a truly random sequence. However, the batteries usually differ with additional tests than the general ones used in most statistical test suites.

As stated by Bill Phelps, Commercial Lead in Booz Allen Consulting [4], increasing the visibility into the supply chain, a good relationship with the suppliers and planning remediation strategies against breaches can help the manufacturer mitigate supply chain

attacks. However, in the supply chain there might be imposed attacks by an adversary after it leaves the manufacturer. These attacks could even happen physically towards the RNG creating external bias to the RNGs entropy source. Such as with thermal noise RNGs where noise-based attacks can be executed to create bias [5].

An adversary may be interested in implementing bias to the RNG, while still making sure that it will be able to pass the most common batteries of statistical tests performed by manufacturers at the point of manufacture, and by the end-customer at the point of usage. However, we can't safely assume that the customer will be able to perform these software tests on their own as it may require some technical knowledge of RNGs. They can be time consuming to perform and require some background information of the statistical tests in order to understand the results as well. Can we introduce another layer of lightweight statistical tests that may remove the trivial steps for the consumer by implementing on-chip hardware statistical tests?

On-chip hardware statistical tests could be an addition, but maybe not alternative, to the software-based lightweight statistical test batteries that exist today. As Hoţoleanu et al. [6] notes, during their experiment implementing the NIST SP800-22 statistical tests onto a FPGA board. They found that only 8 of the 14[1] tests were suitable for lightweight hardware use. The reason why we want a lightweight statistical implementation of such tests is to reduce the time cost of running the tests and the trivial steps to perform RNG testing for the end-customer.

By experimenting with the same techniques used by Hoţoleanu et al. [6] to implement lightweight statistical tests it might be possible to add an additional layer of trust between the customer and the RNG. However, it is uncertain as to if all software tests that have been created as of now is able to be processed solely on hardware without any software input.

[1] 14 out of the 16 total tests in the NIST SP800-22 were recommended by NIST at the time of their experiment [6]

Bibliography

[1] C. M. V. Program, "NIST," 2020. [Online]. Available: https://csrc.nist.gov/Projects/fips-140-3-development. [Accessed 20 November 2020].

[2] D. Hurley-Smith, C. Patsakis and J. Hernandez-Castro, "On the unbearable lightness of FIPS 140-2 randomness tests," *IEEE,* 2020.

[3] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *NIST,* vol. 1a, no. SP800-22, 2010.

[4] B. Phelps, "boozallen.com," 2019. [Online]. Available: https://www.boozallen.com/c/insight/blog/3-ways-to-prevent-supply-chain-attacks.html. [Accessed 20 November 2020].

[5] J. Brown, J. Fu Zhang, B. Zhou, M. Mehedi, P. Freitas, J. Marsland and Z. Ji, "Random-telegraph-noise-enabled true random number generator for hardware security," *Scientific Reports - Nature,* vol. 10, no. 17210, 2020.

[6] D. Hoţoleanu, O. Creţ, A. Suciu, T. Gyorfi and L. Văcariu, "Real-Time Testing of True Random Number Generators Through Dynamic Reconfiguration," *2010 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools,* pp. 247-250, 2010.