WHITE PAPER

Random Number Generators

The building blocks of cybersecurity lie in random numbers — and not just for encryption, but also for authentication, signing, key wrapping, one-time codes, nonces, and more.

> Too often, security professionals have had to make tradeoffs between quality and quantity regarding the random numbers used. In this paper, we review the risks associated with such choices, and discuss new alternatives that address the issue.

Why Random Numbers Are Important in Data Security

Given the extent to which random numbers are used to secure our data, it will come as no surprise that the performance and characteristics of random number generators have a strong impact on security. To put it simply, attackers don't crack encryption, they steal or guess keys. Poor quality or insufficient quantity of random numbers have the effect of making that much easier, reducing security to well below its designed level and making the overall system vulnerable.

Furthermore, as computing power increases, so too will the importance of strong, reliable random number generation, particularly with a growing number of practical applications in quantum computing.

Data Security Breaches Associated with Random Numbers

Serious security issues have resulted from poor random number generators (RNGs). Some examples are given below:

- In Taiwan, weak random numbers used for the nation's digital ID system enabled attackers to carry out identity theft. [1]
- Public keys generated in the RSA security protocol were based on improperly generated random numbers, allowing simple factoring of keys, and therefore access to the protected data. [2]
- In 2013, the RSA crypto library was discovered to have a backdoor in its PRNG, believed to allow the NSA to access those numbers to aid in decrypting protected data. [3]
- The Java platform still relies on a linear congruential generator (LCG) for its PRNG, long known to be of low quality. [4]
- Weak random numbers, along with a weakness in the authentication protocol, allowed any key in a cryptographic RFID system to be found in a matter of seconds. [5]
- The RANDU algorithm, used in mainframe computers for decades was seriously flawed, but its failings went undetected for just as long.

This list isn't exhaustive — there are many more similar examples — but it shows that poor quality random numbers present issues ranging from obvious and known to unnoticed and unknown.

The Power Lies in Entropy

Some of the important parameters to look for in a random number generator are entropy density and throughput. Entropy is a measure of the randomness of the data, while throughput a measure of the quantity of random delivered. For a given throughput, lower entropy will result in keys that are less random, making them more vulnerable to hacking. Similarly, the throughput for a given entropy density determines how much high quality random can be delivered over a certain time interval. This sets for example the frequency at which keys can be rotated. Some random number generators with seemingly high maximum throughputs and maximum entropy will only deliver those high entropy levels at very low throughputs, resulting in a real security trade-off (and risk).

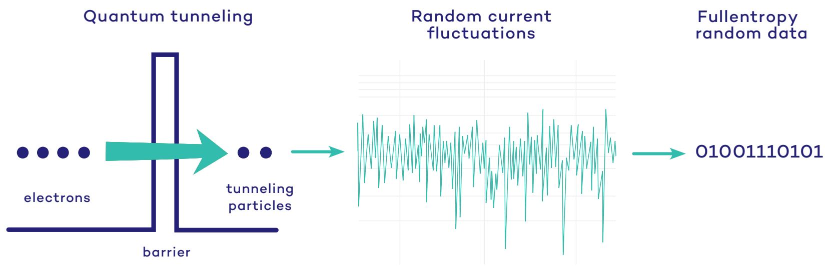
Generation Methods Compared

Methods of random number generation have their strengths and weaknesses, but most approaches struggle to deliver high entropy and high throughput at the same time.

Method	Description	Pros & Cons	Entropy Density
Pseudo-random number generator (PRNG)	Uses algorithms to produce apparently random results, often from short randomization seeds	O Capable of high throughputs X Attacker can derive random number from knowledge of the seed	Fraction of a percentage
Hardware/True random number generator (TRNG)	Measures a physical phenomenon expected to be random	O Can deliver high-quality random X Rate-limited until enough entropy is sampled to meet the demand	Dependent on physical source
Hybrid random number generator	Uses both hardware- and software-derived randomness depending on the read rate needed	O No rate limits X Similar quality to PRNG at high throughputs	Fraction of a percentage at high throughputs
QuintessenceLabs' qStream (quantum random number generator [QRNG])	Measures true random at very high rates directly from a quantum effect	O Full entropy, high throughput; low-latency delivery of high-quality random for all crypto keys	100% entropy at 1 Gbit/sec

Tunneling to Better Random

Access to high quality random at high rates is a crucial part of good data security, with PRNG becoming less viable as computing processing capabilities continue to grow. QuintessenceLabs' delivers quantum random numbers generators (QRNGs) which address both these challenges, while being interoperable with existing security architectures.



qStream QRNG generates full entropy quantum random numbers using quantum tunneling.

The QuintessenceLabs' qStream^{TM} quantum random number generator (QRNG) delivers random numbers by measuring tunneling noise. Quantum tunneling is a well-known quantum phenomenon where charged particles travel ("tunnel") through a barrier that classical (or Newtonian) physics predict they shouldn't be able to cross. Within the QRNG, a voltage is applied to a forward-biased diode junction which serves as the barrier through which some of the charged particles tunnel.

This tunneling within the diode creates random fluctuations in the current flowing through it. Though many particles tunnel through the barrier, the exact number at a given time can't be predicted, yielding a truly random quantum effect and an ideal source of natural entropy. This effect is measured, then digitized and processed to ultimately generate ultra-high-bandwidth random numbers. The full-entropy data is delivered at 1 Gbit/s, suitable for use in any cryptographic application.

QuintessenceLabs' qStream: High Speed Quantum Random Number Generator (QRNG)

Performance

With its 8 Gbit/sec quantum entropy source that delivers 1 Gbit/sec conditioned full-entropy output, the qStream QRNG is the world's leading commercial random number generator. The qStream QRNG has no reliance on deterministic generators nor seeding from external entropy sources. Its RNG output for each client is 100 percent independent, with no shared RNG seeding, no leakage between users, and no ability for users to influence RNG seeding. In addition, the qStream QRNG offers full traceability, from hardware to entropy to random number to key to consumer.

Standards & Testing

The U.S. National Institute of Standards and Technology (NIST)'s recommendations for random number generators are described in Special Publications 800-90A (Deterministic Random Bit Generators), 800-90B (Entropy Sources Used for Random Bit Generators) and 800-90C (Recommendation for Random Bit Generator Constructions - Draft). The qStream QRNG meets all the relevant requirements in these documents.

Perfect random number generators produce "unlikely" sequences of random numbers at exactly the right average rate. Therefore, testing an RNG is quite subtle.

Dieharder is an RNG testing suite, designed to push weak generators to an unambiguous failure. Results from the DieHarder test for the qStream QRNG are shown below:

```
#===========================================================================================#
#        dieharder version 3.31.0 Copyright 2003 Robert G. Brown                            #
#===========================================================================================#
   rng_name    |rands/second|   Seed   |
stdin_input_raw|  3.41e+07  |2746456103|
#===========================================================================================#
 Test_name           |Assessment Test_name          |Assessment Test_name        |Assessment
#===========================================================================================#
 diehard_birthdays   |  PASSED   diehard_xdsphere   |  PASSED   gb_minimum_distance |  PASSED
 diehard_operm5      |  PASSED   diehard_squeeze    |  PASSED   rgb_permutations    |  PASSED
 diehard_rank        |  PASSED   diehard_sums       |  PASSED   rgb_lagged_sum      |  PASSED
 diehard_bitstream   |  PASSED   diehard_runs       |  PASSED   rgb_kstest_test     |  PASSED
 diehard_opso        |  PASSED   diehard_craps      |  PASSED   dab_bytedistrib     |  PASSED
 diehard_oqso        |  PASSED   marsaglia_tsang_gcd|  PASSED   dab_dct             |  PASSED
 diehard_dna         |  PASSED   sts_monobit        |  PASSED   dab_filltree        |  PASSED
 diehard_count_1s_str|  PASSED   sts_runs           |  PASSED   dab_monobit2        |  PASSED
 diehard_count_1s_byt|  PASSED   sts_serial         |  PASSED
 diehard_parking_lot |  PASSED   rgb_bitdist        |  PASSED
```

Alongside that is the NIST Statistical Test Suite, one of the most frequently used test batteries. The following are the qStream QRNG's results:

```
----------------------------------------------------------------------
--------
RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING
SEQUENCES
----------------------------------------------------------------------
--------
   generator is <nist_test.output>
----------------------------------------------------------------------
--------
STATISTICAL TEST | PROPORTION     STATISTICAL TEST | PROPORTION
----------------------------------------------------------------------
--------
Frequency        |   10/10        NonOverlappingTemplate |   10/10
BlockFrequency   |   10/10        OverlappingTemplate    |   10/10
CumulativeSums   |   10/10        Universal              |   10/10
Runs             |   10/10        ApproximateEntropy     |   10/10
LongestRun       |   10/10        RandomExcursions       |   5/6
Rank             |   10/10        RandomExcursionsVariant|   6/6
FFT              |   9/10         LinearComplexity       |   10/10
Serial           |   10/10
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - -
- The minimum pass rate for each statistical test with the exception of
the
random excursion (variant) test is approximately = 8 for a
sample size = 10 binary sequences.
- The minimum pass rate for the random excursion (variant) test
is approximately = 5 for a sample size = 6 binary sequences.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - -
```

Other Considerations

This white paper has concentrated on entropy and throughput as two important parts of a random number generator, but other characteristics should be considered. Chiefly, this includes the ability of the RNG to integrate and operate securely within the overall security architecture of an organization — for example, a key manager managing the lifecycle of random objects; interfaces other elements of a security architecture; secure network access to the random; audit trails, and more.

For more information on how QuintessenceLabs performs with respect to all these parameters, visit **quintessencelabs.com** or contact **info@quintessencelabs.com.**

References

1. *Fatal Crypto Flaw in Some Government-Certified Smartcards Makes Forgery a Snap*, Dan Goodin, Ars Technica, 2013-09-16.
2. *Ron was Wrong, Whit is Right*, Arjen K. Lenstra et al., International Association for Cryptologic Research, 2012.
3. *N.S.A. Able to Foil Basic Safeguards of Privacy on Web*, Nicole Perlroth, The New York Times, 2013-09-05.
4. *Recovering Private Keys Generated with Weak PRNGs*, P.-A. Fouque, M. Tibouchi and J.C. Zapalowicz, 14th IMA International Conference on Cryptography and Coding, 2013-12-17.
5. *Reverse-Engineering a Cryptographic RFID Tag*, Karsten Nohl et al., SS'08 Proceedings of the 17th Conference on Security Symposium, 2008-07-28.

©2020 QuintessenceLabs. All rights reserved.