Hardware random number generator

In computing, a **hardware random number generator** (**HRNG**), **true random number generator** (**TRNG**), **non-deterministic random bit generator** (**NRBG**),[1] or **physical random number generator**[2][3] is a device that generates random numbers from a physical process capable of producing entropy (in other words, the device always has access to a physical **entropy source**[1]), unlike the pseudorandom number generator (PRNG, a.k.a. "deterministic random bit generator", DRBG) that utilizes a deterministic algorithm[2] and non-physical nondeterministic random bit generators that do not include hardware dedicated to generation of entropy.[1]


A USB-pluggable hardware true random number generator

Many natural phenomena generate low-level, statistically random "noise" signals, including thermal and shot noise, jitter and metastability of electronic circuits, Brownian motion, and atmospheric noise.[4] Researchers also used the photoelectric effect, involving a beam splitter, other quantum phenomena,[5][6][7][8][9] and even the nuclear decay (due to practical considerations the latter, as well as the atmospheric noise, is not viable).[4] While "classical" (non-quantum) phenomena are not truly random, an unpredictable physical system is usually acceptable as a source of randomness, so the qualifiers "true" and "physical" are used interchangeably.[10]

A hardware random number generator is expected to output near-perfect random numbers ("full entropy").[1] A physical process usually does not have this property, and a practical TRNG typically includes a few blocks:[11]

- a *noise source* that implements the physical process producing the entropy. Usually this process is analog, so a *digitizer* is used to convert the output of the analog source into a binary representation;
- a *conditioner* (randomness extractor) that improves the quality of the random bits;
- *health tests*. TRNGs are mostly used in cryptographical algorithms that get completely broken if the random numbers have low entropy, so the testing functionality is usually included.

Hardware random number generators generally produce only a limited number of random bits per second. In order to increase the available output data rate, they are often used to generate the "seed" for a faster PRNG. DRBG also helps with the noise source "anonymization" (whitening out the noise source identifying characteristics) and entropy extraction. With a proper DRBG algorithm selected (cryptographically secure pseudorandom number generator, CSPRNG), the combination can satisfy the requirements of Federal Information Processing Standards and Common Criteria standards.[12]

Uses

Hardware random number generators can be used in any application that needs randomness. However, in many scientific applications additional cost and complexity of a TRNG (when compared with pseudo random number generators) provide no meaningful benefits. TRNGs have additional drawbacks for data science and statistical applications: impossibility to re-run a series of numbers unless they are stored, reliance on an analog physical entity can obscure the failure of the source. The TRNGs therefore are primarily used in the applications where their unpredictability and the impossibility to re-run the sequence of numbers are crucial to the success of the implementation: in cryptography and gambling machines.[13]

Cryptography

The major use for hardware random number generators is in the field of data encryption, for example to create random cryptographic keys and nonces needed to encrypt and sign data. In addition to randomness, there are at least two additional requirements imposed by the cryptographic applications:[14]

1. forward secrecy guarantees that the knowledge of the past output and internal state of the device should not enable the attacker to predict future data;
2. backward secrecy protects the "opposite direction": knowledge of the output and internal state in the future should not divulge the preceding data.

A typical way to fulfill these requirements is to use a TRNG to seed a cryptographically secure pseudorandom number generator.[15]

History

Physical devices were used to generate random numbers for thousands of years, primarily for gambling. Dice in particular have been known for more than 5000 years (found on locations in modern Iraq and Iran), and flipping a coin (thus producing a random bit) dates at least to the times of ancient Rome.[16]

The first documented use of a physical random number generator for scientific purposes was by Francis Galton (1890).[17] He devised a way to sample a probability distribution using a common gambling dice. In addition to the top digit, Galton also looked at the face of a dice closest to him, thus creating 6*4 = 24 outcomes (about 4.6 bits of randomness).[16]

Kendall and Babington-Smith (1938)[18] used a fast-rotating 10-sector disk that was illuminated by periodic bursts of light. The sampling was done by a human who wrote the number under the light beam onto a pad. The device was utilized to produce a 100,000-digit random number table (at the time such tables were used for statistical experiments, like PRNG nowadays).[16]

On 29 April 1947, the RAND Corporation began generating random digits with an "electronic roulette wheel", consisting of a random frequency pulse source of about 100,000 pulses per second gated once per second with a constant frequency pulse and fed into a five-bit binary counter. Douglas Aircraft built the equipment, implementing Cecil Hasting's suggestion (RAND P-113)[19] for a noise source (most likely the well known behavior of the 6D4 miniature gas thyratron tube, when placed in a magnetic field[20]). Twenty of the 32 possible counter values were mapped onto the 10 decimal digits and the other 12 counter values were discarded.[21] The results of a long run from the RAND machine, filtered and tested, were converted into a table, which originally existed only as a deck of punched cards, but was later published in 1955 as a book, 50 rows of 50 digits on each page[16] (*A Million Random Digits with 100,000 Normal Deviates*). The RAND table was a significant breakthrough in delivering random numbers because such a large and carefully prepared table had never before been available. It has been a useful source for simulations, modeling, and for deriving the arbitrary constants in cryptographic algorithms to demonstrate that the constants had not been selected maliciously ("nothing up my sleeve numbers"). [22]

Since the early 1950s, research into TRNGs has been highly active, with thousands of research works published and about 2000 patents granted by 2017.[16]

Physical phenomena with random properties

Multiple different TRNG designs were proposed over time with a large variety of noise sources and digitization techniques ("harvesting"). However, practical considerations (size, power, cost, performance, robustness) dictate the following desirable traits:[23]

- use of a commonly available inexpensive silicon process;
- exclusive use of digital design techniques. This allows an easier system-on-chip integration and enables the use of FPGAs;
- compact and low-power design. This discourages use of analog components (e.g., amplifiers);
- mathematical justification of the entropy collection mechanisms.

Stipčević & Koç in 2014 classified the physical phenomena used to implement TRNG into four groups:[3]

- electrical noise;
- free-running oscillators;
- chaos;
- quantum effects.

Electrical noise-based RNG

Noise-based RNGs generally follow the same outline: the source of a noise generator is fed into a comparator. If the voltage is above threshold, the comparator output is 1, otherwise 0. The random bit value is latched using a flip-flop. Sources of noise vary and include:[24]

- Johnson–Nyquist noise ("thermal noise");
- Zener noise;
- avalanche breakdown.

The drawbacks of using noise sources for an RNG design are:[25]

- noise levels are hard to control, they vary with environmental changes and device-to-device;
- calibration processes needed to ensure a guaranteed amount of entropy are time-consuming;
- noise levels are typically low, thus the design requires power-hungry amplifiers. The sensitivity of amplifier inputs enables manipulation by an attacker;
- circuitry located nearby generates a lot of non-random noise thus lowering the entropy;
- a proof of randomness is near-impossible as multiple interacting physical processes are involved.[26]

Chaos-based RNG

The idea of chaos-based noise stems from the use of a complex system that is hard to characterize by observing its behavior over time. For example, lasers can be put into (undesirable in other applications) chaos mode with chaotically fluctuating power, with power detected using a photodiode and sampled by a comparator. The design can be quite small, as all photonics elements can be integrated on-chip. Stipčević & Koç characterize this technique as "most objectionable", mostly due to the fact that chaotic behavior is usually controlled by a differential equation and no new randomness is introduced, thus there is a possibility of the chaos-based TRNG producing a limited subset of possible output strings.[27]

Free-running oscillators-based RNG

The TRNGs based on a free-running oscillator (FRO) typically utilize one or more ring oscillators (ROs), outputs of which are sampled using yet another clock. Since inverters forming the RO can be thought of as amplifiers with a very large gain, an FRO output exhibits very fast oscillations in phase and frequency domains. The FRO-based TRNGs are very popular due to their use of the standard digital logic despite issues with randomness proofs and chip-to-chip variability.[27]

Quantum-based RNG

Quantum random number generation technology is well established with 8 commercial **quantum random number generator** (**QRNG**) products offered before 2017.[28]

Herrero-Collantes & Garcia-Escartin list the following stochastic processes as "quantum":

- nuclear decay historically was the earliest quantum method used since the 1960s owing its popularity to the availability of Geiger counters and calibrated radiation sources. The entropy harvesting was done using an event counter that was periodically sampled or a time counter that was sampled at the time of the event. Similar designs were utilized in the 1950s to generate random noise in analog computers. The major drawbacks were radiation safety concerns, low bit rates, and non-uniform distribution;[29]
- shot noise, a quantum mechanical noise source found in electronic circuits, while technically a quantum effect, is hard to isolate from the thermal noise, so, with few exceptions, noise sources utilizing it are only partially quantum and are usually classified as "classical";[30]
- quantum optics:
  - *branching path generator* using a beamsplitter so that a photon from a single-photon source randomly takes one of the two paths and sensed by one of the two single-photon detectors thus generating a random bit;[31]
  - *time of arrival generators* and *photon counting generators* use a weak photon source, with the entropy harvested similarly to the case of radioactive decay;[32]
  - *attenuated pulse generators* are a generalization (simplifying the equipment) of the above methods that allows more than one photon in the system at a time;[33]
  - *vacuum fluctuations generators* use a laser homodyne detection to probe the changes in the vacuum state;[34]
  - *laser phase noise generators* use the phase noise on the output of a single spatial mode laser that is converted to amplitude using an unbalanced Mach-Zehnder interferometer. The noise is sampled by a photodetector;[35]
  - amplified spontaneous emission generators use spontaneous light emission present in the optical amplifiers as a source of noise;[36]
  - *Raman scattering generators* extract entropy from the interaction of photons with the solid-state materials;[37]
  - *optical parametric oscillator generators* use the spontaneous parametric down-conversion leading to binary phase state selection in a degenerate optical parametric oscillator;[38]

To reduce costs and increase robustness of quantum random number generators,[39] online services have been implemented.[28]

A plurality of quantum random number generators designs[40] are inherently untestable and thus can be manipulated by adversaries. Mannalath et al. call these designs "trusted" in a sense that they can only operate in a fully controlled, trusted environment.[41]

Performance test

The failure of a TRNG can be quite complex and subtle, necessitating validation of not just the results (the output bit stream), but of the unpredictability of the entropy source.[10] Hardware random number generators should be constantly monitored for proper operation to protect against the entropy source degradation due to natural causes and deliberate attacks. FIPS Pub 140-2 and NIST Special Publication 800-90B[42] define tests which can be used for this.

The minimal set of real-time tests mandated by the certification bodies is not large; for example, NIST in SP 800-90B requires just two *continuous health tests*:[43]

1. *repetition count test* checks that the sequences of identical digits are not too long, for a (typical) case of a TRNG that digitizes one bit at a time, this means not having long strings of either 0s or 1s;
2. *adaptive proportion test* verifies that any random digit does not occur too frequently in the data stream (low bias). For bit-oriented entropy sources that means that the count of 1s and 0s in the bit stream is approximately the same.

Attacks

Just as with other components of a cryptography system, a cryptographic random number generator should be designed to resist certain attacks. Defending against these attacks is difficult without a hardware entropy source.

The physical processes in HRNG introduce new attack surfaces. For example, a free-running oscillator-based TRNG can be attacked using a frequency injection.[44]

Estimating entropy

There are mathematical techniques for estimating the entropy of a sequence of symbols. None are so reliable that their estimates can be fully relied upon; there are always assumptions which may be very difficult to confirm. These are useful for determining if there is enough entropy in a seed pool, for example, but they cannot, in general, distinguish between a true random source and a pseudorandom generator. This problem is avoided by the conservative use of hardware entropy sources.

See also

- AN/CYZ-9
- Bell test experiments
- /dev/random
- ERNIE
- Lavarand (a hardware random number generator based on movement of the floating material in lava lamps)
- List of random number generators
- Lottery machine
- RDRAND
- Trusted Platform Module

References

1. Turan et al. 2018, p. 64.
2. Schindler 2009, p. 7.
3. Stipčević & Koç 2014, p. 279.
4. Sunar 2009, p. 56.
5. Herrero-Collantes & Garcia-Escartin 2017, p. 8.
6. Jacak, Marcin M.; Jóźwiak, Piotr; Niemczuk, Jakub; Jacak, Janusz E. (2021). "Quantum generators of random numbers" (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8352985). *Scientific Reports*. **11** (1): 16108. Bibcode:2021NatSR..1116108J (https://ui.adsabs.harvard.edu/abs/2021NatSR..1116108J). doi:10.1038/s41598-021-95388-7 (https://doi.org/10.1038%2Fs41598-021-95388-7). PMC 8352985 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8352985). PMID 34373502 (https://pubmed.ncbi.nlm.nih.gov/34373502).
7. Ma, Xiongfeng; Yuan, Xiao; Cao, Zhu; Qi, Bing; Zhang, Zhen (2016). "Quantum random number generation" (https://doi.org/10.1038%2Fnpjqi.2016.21). *npj Quantum Information*. **2** (1): 16021. arXiv:1510.08957 (https://arxiv.org/abs/1510.08957). Bibcode:2016npjQI...216021M (https://ui.adsabs.harvard.edu/abs/2016npjQI...216021M). doi:10.1038/npjqi.2016.21 (https://doi.org/10.1038%2Fnpjqi.2016.21).
8. Kollmitzer, Christian; Petscharnig, Stefan; Suda, Martin; Mehic, Miralem (2020). "Quantum Random Number Generation" (https://link.springer.com/chapter/10.1007/978-3-319-72596-3_2). *Quantum Random Number Generation: Theory and Practice*. Springer International Publishing. pp. 11–34. doi:10.1007/978-3-319-72596-3_2 (https://doi.org/10.1007%2F978-3-319-72596-3_2). ISBN 978-3-319-72596-3.
9. Mannalath, Mishra & Pathak 2023.
10. Herrero-Collantes & Garcia-Escartin 2017, p. 4.
11. Turan et al. 2018, p. 6.
12. Saarinen, Newell & Marshall 2020.
13. Templ 2016, p. 90.
14. Herrero-Collantes & Garcia-Escartin 2017, p. 6.
15. Herrero-Collantes & Garcia-Escartin 2017, p. 7.
16. L'Ecuyer 2017.
17. Galton, Francis (1890). "Dice for statistical experiments" (http://galton.org/essays/1890-1899/galton-1890-dice.pdf) (PDF). *Nature*. **42** (1070): 13–14. Bibcode:1890Natur..42...13G (https://ui.adsabs.harvard.edu/abs/1890Natur..42...13G). doi:10.1038/042013a0 (https://doi.org/10.1038%2F042013a0). S2CID 4038609 (https://api.semanticscholar.org/CorpusID:4038609). Archived (https://web.archive.org/web/20160304000756/http://galton.org/essays/1890-1899/galton-1890-dice.pdf) (PDF) from the original on 4 March 2016. Retrieved 14 May 2014.
18. Kendall, M. G., and B. Babington-Smith. 1938. “Randomness and other random sampling numbers”. Journal of the Royal Statistical Society 101:147–166.
19. Brown, George W. (January 1949), *P-113* (http://www.rand.org/pubs/papers/P113), Papers, Rand Corporation, archived (https://web.archive.org/web/20070605075201/http://www.rand.org/pubs/papers/P113/) from the original on 2007-06-05, retrieved 2009-05-10.
20. Cobine, Curry (1947), "Electrical Noise Generators", *Proceedings of the I.R.E.* (September 1947): 875–9
21. *Monograph report* (http://www.rand.org/pubs/monograph_reports/MR1418/index2.html), Rand Corporation, January 2001, archived (https://web.archive.org/web/20180415035650/https://www.rand.org/pubs/monograph_reports/MR1418/index2.html) from the original on 2018-04-15, retrieved 2009-01-29.
22. Schneier, Bruce (1995-11-01). "Other Stream Ciphers and Real Random-Sequence Generators". *Applied Cryptography* (Second ed.). John Wiley & Sons, Inc. p. 423. ISBN 978-0-471-11709-4.
23. Sunar 2009, p. 57.
24. Stipčević & Koç 2014, pp. 279–280.
25. Stipčević & Koç 2014, p. 280.
26. Stipčević & Koç 2014, p. 286.
27. Stipčević & Koç 2014, pp. 288–289.
28. Herrero-Collantes & Garcia-Escartin 2017, p. 2.
29. Herrero-Collantes & Garcia-Escartin 2017, pp. 10–13.
30. Herrero-Collantes & Garcia-Escartin 2017, pp. 13–14.
31. Herrero-Collantes & Garcia-Escartin 2017, p. 15.
32. Herrero-Collantes & Garcia-Escartin 2017, p. 17.
33. Herrero-Collantes & Garcia-Escartin 2017, p. 20.
34. Herrero-Collantes & Garcia-Escartin 2017, pp. 20–21.
35. Herrero-Collantes & Garcia-Escartin 2017, pp. 21–22.
36. Herrero-Collantes & Garcia-Escartin 2017, pp. 23–24.

37. Herrero-Collantes & Garcia-Escartin 2017, pp. 24–25.
38. Herrero-Collantes & Garcia-Escartin 2017, pp. 27–28.
39. Huang, Leilei; Zhou, Hongyi; Feng, Kai; Xie, Chongjin (2021-07-07). "Quantum random number cloud platform" (https://doi.org/10.1038%2Fs41534-021-00442-x). *npj Quantum Information*. **7** (1). Springer Science and Business Media LLC: 107. Bibcode:2021npjQI...7..107H (https://ui.adsabs.harvard.edu/abs/2021npjQI...7..107H). doi:10.1038/s41534-021-00442-x (https://doi.org/10.1038%2Fs41534-021-00442-x). ISSN 2056-6387 (https://search.worldcat.org/issn/2056-6387).
40. Mannalath, Mishra & Pathak 2023, p. 4.
41. Mannalath, Mishra & Pathak 2023, p. 9.
42. Turan et al. 2018.
43. Turan et al. 2018, pp. 25–27.
44. Markettos, A. Theodore; Moore, Simon W. (2009). "The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators". *Lecture Notes in Computer Science* (https://www.academia.edu/download/69622562/978-3-642-04138-9_23.pdf) (PDF). Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 317–331. doi:10.1007/978-3-642-04138-9_23 (https://doi.org/10.1007%2F978-3-642-04138-9_23). ISBN 978-3-642-04137-2. ISSN 0302-9743 (https://search.worldcat.org/issn/0302-9743).

Sources

- Turan, Meltem Sönmez; Barker, Elaine; Kelsey, John; McKay, Kerry A; Baish, Mary L; Boyle, Mike (2018). NIST SP800-90B: Recommendation for the entropy sources used for random bit generation (Report). Gaithersburg, MD: National Institute of Standards and Technology. doi:10.6028/nist.sp.800-90b (https://doi.org/10.6028%2Fnist.sp.800-90b).
- Templ, M. (2016). *Simulation for Data Science with R* (https://books.google.com/books?id=EAJwDQAAQBAJ&pg=PA90). Packt Publishing. ISBN 978-1-78588-587-7. Retrieved 2023-08-07.
- Saarinen, Markku-Juhani O.; Newell, G. Richard; Marshall, Ben (2020-11-09). *Building a Modern TRNG: An Entropy Source Interface for RISC-V* (https://web.archive.org/web/20210316160110/https://eprint.iacr.org/2020/866.pdf) (PDF). New York, NY, USA: ACM. doi:10.1145/3411504.3421212 (https://doi.org/10.1145%2F3411504.3421212). Archived from the original on 2021-03-16. Retrieved 2023-09-09.
- Schindler, Werner (2009). "Random Number Generators for Cryptographic Applications". *Cryptographic Engineering*. Boston, MA: Springer US. pp. 5–23. doi:10.1007/978-0-387-71817-0_2 (https://doi.org/10.1007%2F978-0-387-71817-0_2). ISBN 978-0-387-71816-3.
- Sunar, Berk (2009). "True Random Number Generators for Cryptography". *Cryptographic Engineering*. Boston, MA: Springer US. pp. 55–73. doi:10.1007/978-0-387-71817-0_4 (https://doi.org/10.1007%2F978-0-387-71817-0_4). ISBN 978-0-387-71816-3.
- L'Ecuyer, Pierre (2017). *History of uniform random number generation* (https://inria.hal.science/hal-01561551/file/wsc17rng-history-report.pdf) (PDF). 2017 Winter Simulation Conference (WSC). Las Vegas, NV, USA: IEEE. doi:10.1109/wsc.2017.8247790 (https://doi.org/10.1109%2Fwsc.2017.8247790). ISBN 978-1-5386-3428-8. ISSN 1558-4305 (https://search.worldcat.org/issn/1558-4305).
- Stipčević, Mario; Koç, Çetin Kaya (2014). "True Random Number Generators". *Open Problems in Mathematics and Computational Science* (https://cetinkayakoc.net/docs/b08.pdf) (PDF). Cham: Springer International Publishing. pp. 275–315. doi:10.1007/978-3-319-10683-0_12 (https://doi.org/10.1007%2F978-3-319-10683-0_12). ISBN 978-3-319-10682-3.
- Herrero-Collantes, Miguel; Garcia-Escartin, Juan Carlos (2017-02-22). "Quantum random number generators". *Reviews of Modern Physics*. **89** (1). American Physical Society (APS): 015004. arXiv:1604.03304 (https://arxiv.org/abs/1604.03304). Bibcode:2017RvMP...89a5004H (https://ui.adsabs.harvard.edu/abs/2017RvMP...89a5004H). doi:10.1103/revmodphys.89.015004 (https://doi.org/10.1103%2Frevmodphys.89.015004). ISSN 0034-6861 (https://search.worldcat.org/issn/0034-6861).
- *Quantum Random Number Generation: Theory and Practice*. Quantum Science and Technology. Springer Cham. 2020. doi:10.1007/978-3-319-72596-3 (https://doi.org/10.1007%2F978-3-319-72596-3). ISBN 978-3-319-72596-3.
- Mannalath, Vaisakh; Mishra, Sandeep; Pathak, Anirban (2023). "A Comprehensive Review of Quantum Random Number Generators: Concepts, Classification and the Origin of Randomness" (https://doi.org/10.1007%2Fs11128-023-04175-y). *Quantum Information Processing*. **22** (12): 439. arXiv:2203.00261 (https://arxiv.org/abs/2203.00261). Bibcode:2023QuIP...22..439M (https://ui.adsabs.harvard.edu/abs/2023QuIP...22..439M). doi:10.1007/s11128-023-04175-y (https://doi.org/10.1007%2Fs11128-023-04175-y).

General references

- Brown, George W (June 1949), *History of Rand's Million Digits* (http://www.rand.org/pubs/papers/P113), papers, RAND Corporation, archived (https://web.archive.org/web/20070605075201/http://www.rand.org/pubs/papers/P113/) from the original on 2007-06-05, retrieved 2009-05-10
- Brown, Bernice (October 1948), *Some Tests of the Randomness of a Million Digits* (http://www.rand.org/pubs/papers/P44), Papers, RAND Corporation, archived (https://web.archive.org/web/20070605075822/http://www.rand.org/pubs/papers/P44/) from the original on 2007-06-05, retrieved 2009-05-10
- "Tube type 6D4", *Electron Tube Data handbook*, Sylvania, 1957
- *A Million Random Digits with 100,000 Normal Deviates* (http://www.rand.org/publications/classics/randomdigits/), RAND Corporation, January 2001, archived (https://web.archive.org/web/20021216191633/http://www.rand.org/publications/classics/randomdigits/) from the original on 2002-12-16, retrieved 2002-12-22.
- Galton, Francis (1890), "Dice for statistical experiments" (http://www.mugu.com/galton/statistician.html), *Nature*, **42** (1070): 13–4, Bibcode:1890Natur..42...13G (https://ui.adsabs.harvard.edu/abs/1890Natur..42...13G), doi:10.1038/042013a0 (https://doi.org/10.1038%2F042013a0), archived (https://web.archive.org/web/20040404063102/http://www.mugu.com/galton/statistician.html) from the original on 2004-04-04, retrieved 2004-03-28
- *Randomness and Genuine Random Number Generator With Self-testing Functions* (https://web.archive.org/web/20180301104756/https://geant4.web.cern.ch/geant4/results/papers/QMD-MC2010.pdf) (PDF), Japan: LE Tech RNG, archived from the original (https://geant4.web.cern.ch/geant4/results/papers/QMD-MC2010.pdf) (PDF) on 2018-03-01, retrieved 2015-04-20
- D. Eastlake, 3rd; J. Schiller; S. Crocker (June 2005). *Randomness Requirements for Security* (https://datatracker.ietf.org/doc/html/rfc4086). doi:10.17487/RFC4086 (https://doi.org/10.17487%2FRFC4086). BCP 106. RFC 4086 (https://datatracker.ietf.org/doc/html/rfc4086). *Best Current Practice 106.* Obsoletes RFC 1750 (https://datatracker.ietf.org/doc/html/rfc1750).

External links

- *The Intel Random Number Generator* (http://www.cryptography.com/public/pdf/IntelRNG.pdf) (PDF), Intel, 22 April 1999.
- ProtegoST SG100 (https://www.protegost.com/), ProtegoST, "Hardware Random Number Generator "Based on quantum physics random number source from a zener diode".