Random Number Generation from Pulsars

Hayder Tirmazi

City College of New York.

Contributing authors: hayder.research@gmail.com;

Abstract

Pulsars exhibit signals with precise inter-arrival times that are on the order of milliseconds to seconds, depending on the individual pulsar. There are subtle variations in the timing of pulsar signals. We show that these variations can serve as a natural entropy source for the creation of Random Number Generators (RNGs). We also explore the effects of using randomness extractors to increase the entropy of random bits extracted from Pulsar timing data. To evaluate the quality of the Pulsar RNG, we model its entropy as a k-source and use well-known cryptographic results to show its closeness to a theoretically ideal uniformly random source. To remain consistent with prior work, we also show that the Pulsar RNG passes well-known statistical tests such as the NIST test suite.

Keywords: cryptographic randomness, random number generation, trngs, astronomy

1 Introduction

Random number generators (RNGs) are a fundamental part of modern cryptography [1]. They can be used to implement provably secure secret-key encryption schemes [1, 2], digital signature schemes [1], and the key generation step of public-key encryption schemes such as RSA [3] and [4]. True Random Number Generators (TRNGS) use noise in physical processes as a source of randomness. As an example, Intel's TRNG uses Johnson noise in resistors [5]. Pseudo-Random Number Generators (PRNGs) are initialized with a *seed* and use algorithms to produce numbers that seem random to adversaries that do not know the seed and are restricted to performing all their computations in probabilistic polynomial time [2]. The initial seed of a PRNG may be derived from a TRNG. Prior work on extracting randomness from astrophysical sources includes, in chronological order, hot pixels in astronomical imaging [6], radio astronomy

signal data noise [7], cosmic microwave background radiation spectra [8], cosmic photon arrival times [9], and intrinsic flux density distribution of single pulsars [10].

Pulsar timing variations provide an alternative entropy source that is structured yet unpredictable. To the best of our knowledge, this paper is the first to investigate the variation in inter-arrival times of pulsar signals as a novel entropy source for cryptographic random number generation. Prior work on random number generation from astrophysical sources notably including [10] and [6] has relied primarily on black-box statistical testing to evaluate randomness quality. There are well-known concerns [11] with relying solely on such statistical tests without a proper theoretical analysis of the entropy source. In fact, the statistical tests endorsed by NIST can be passed even by weak PRNGs [11]. Unlike prior work, our work also includes a theoretical analysis using known cryptographic techniques to complement our empirical findings.

The rest of this paper is structured as follows. In Section 2, we create a Pulsar RNG from observational data from two sources: the North American Nanohertz Observatory for Gravitational Waves (NANOGrav) [12] and the European Pulsar Timing Array (EPTA) [13]. Section 3 evaluates the Pulsar RNG using a cryptographic analysis and statistical tests. Section 4 provides discussion relevant to the viability of the Pulsar RNG.

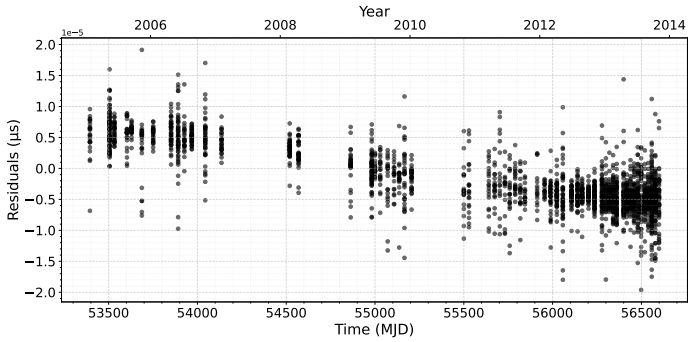
2 Generating Random Bits

We use measurement data from two pulsars, PSR J0030+0451 and PSR J1918-0642. These two pulsars are present in both the NANOGrav 9-year dataset release [5] and the EPTA DR2 dataset release [13]. Our Pulsar RNG extracts timing residuals from these datasets using PINT [14] v1.1.1. Let L be the list of pulsar residuals where L_i is the ith element. We first normalize the residual values to create list N in the usual way ($N_i = \frac{L_i - \min(L)}{\max(L) - \min(L)}$). We then investigate three quantification techniques on the list N to convert it to a list of random bits R.

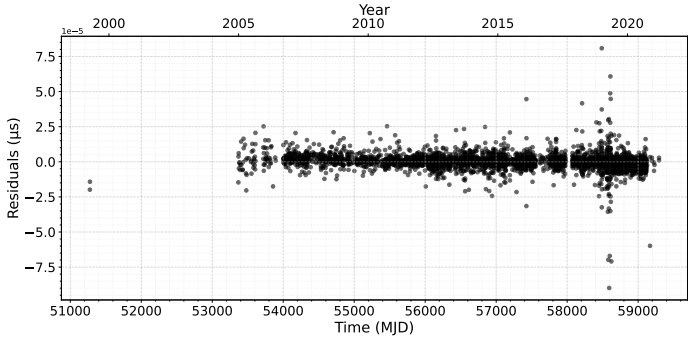
1. A simple threshold: $R_i = 1$ if $N_i \geq 0.5$, otherwise $R_i = 0$
2. 8-bit Gray coding [15]
3. Using the 8-bit Gray coded value as a seed for a SHA-512 hash [16]

Figure 2 shows the measured dataset entropy in bits per byte of these three quantification methods. Note that by *entropy* throughout this paper we mean information entropy, also known as Shannon entropy. We measure all dataset entropy results in this paper using the `ent` tool [17].

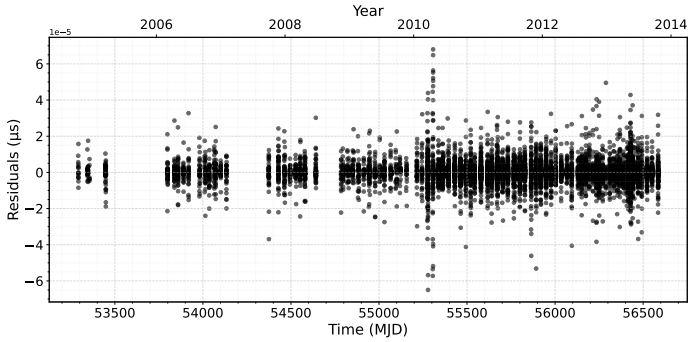
Using threshold as a quantification method requires a careful choice of where to put the threshold based on each distribution. Our method of uniformly using $\tau = 0.5$ as the threshold provides vastly different results for, as an example, the PSR J1918-0642 data on the EPTA dataset as opposed to the NANOGrav dataset. This is due to $\tau = 0.5$ not providing an equal direction of the EPTA data. This can be verified in Figure 3 which shows normalized residuals (N) for both datasets. Notice that while the points on the NANOGrav data are roughly equally divided by a cutoff line at 0.5 (marked by a dotted line in the figure), most points in the EPTA dataset are below 0.5.



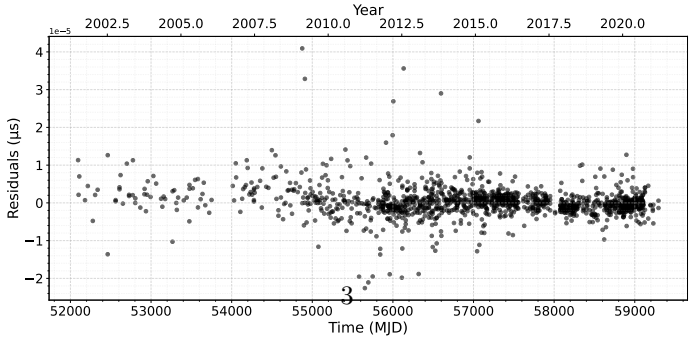
(a) J0030+0451 NANOGrav



(b) J0030+0451 EPTA



(c) J1918-0642 NANOGrav



(d) J1918-0642 EPTA

Fig. 1: Timing Variations in μs for the J0030+0451 and J1918-0642 pulsars with Modified Julian Date (MJD) and Year plotted on the x axis

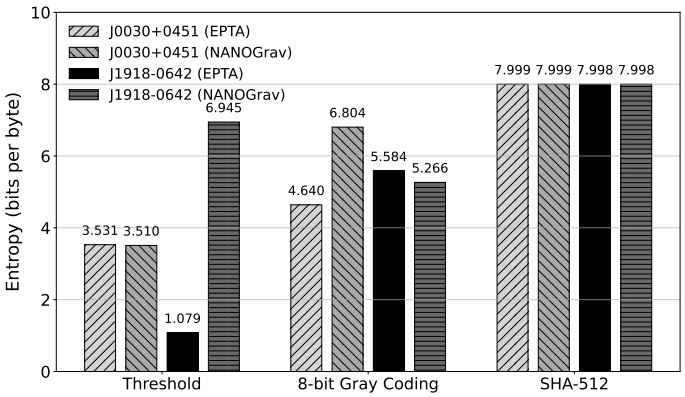


Fig. 2: Entropy of Different Quantification Methods for 2 Pulsars across EPTA and NANOGrav data

We next investigate the effect of three different randomness extractors on our results. Randomness Extractors are functions that take as input 1) a comparatively small uniformly-random seed and 2) a comparatively weak entropy source, for example, radioactive decay [18] or in our case Pulsar timing variation. Randomness Extractors output random bits that appear to computationally bound adversaries as being independent from the input entropy source and uniformly randomly distributed. Note that prior astrophysics-based RNG papers including [6] refer to randomness extractors as debiasing or deskewing algorithms. We test two simple *ad hoc* randomness extractors, XOR-ing several subsequent bits [19] and [20]. We also test a Randomness extractor based on SHAKE-256 from the SHA-3 family of cryptographic hash functions. Figure 4 shows our results. While using a cryptographic hash yields the highest entropy, it is interesting to note that even an ad-hoc random extractor like Von Neumann provides considerable entropy gains.

3 Evaluation

A strict mathematical proof of absolute randomness is considered impossible [19]. To analyze TRNGs, we must rely on assumptions based on the fundamental postulates of physics [19] in combination with our mathematical analysis. We define randomness extractors and k-sources using standard cryptographic definitions (See A for details). We use our definitions to show the suitability of Pulsar RNG under a reasonable physical assumption. We assume that pulsar timing variations exhibit non-trivial entropy and can be modeled as a k-source (Assumption 1). From a theoretical standpoint, this assumption aligns with existing stochastic models [21] of pulsar timing variations due to non-deterministic phenomena such as glitches [22], as well as the presence of Gravitational Waves [23].

We also empirically verify our assumption based on Pulsar data from NANOGrav and EPTA. We show our empirical results in Table 1. We generate binary arrays from 10 different Pulsars, 5 in the NANOGrav dataset and 5 in the EPTA dataset

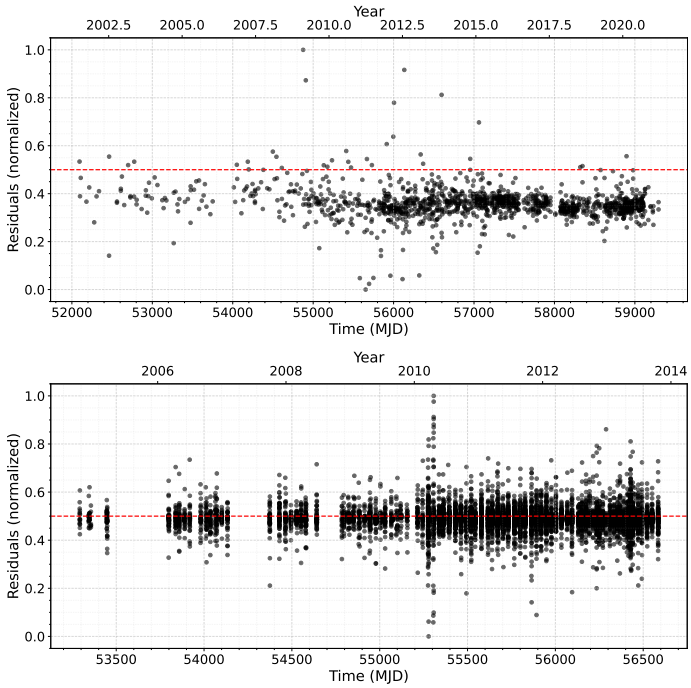


Fig. 3: Normalized PSR J1918-0642 residuals on EPTA data (above) and NANOGrav data (below).

respectively. Then we measure the min-entropy (Definition 3) of generated binary arrays in units of bits per bit. By a non-trivial min-entropy, we mean a min-entropy value significantly larger than 0. By definition, the min-entropy over a binary array will be in the range $[0, 1]$ in bits per bit. We rely on the 8-bit Gray coding method for quantification that we discussed in Section 2.

3.1 Cryptographic Guarantees

We show that our Pulsar RNG satisfies the conditions of a strong extractor under the Leftover Hash Lemma. Randomness extractors are cryptographic primitives that can transform an entropy source with bias into a (in practice) uniformly random distribution. The Leftover Hash Lemma formally proves that a universal hashing family can extract nearly uniform bits from a k-source. For the hash function that performs this debiasing in Pulsar RNG, we use SHAKE-256 from the SHA-3 family of cryptographic hash functions. The formal proof is provided in A. Informally, this result implies that random bits generated by Pulsar RNGs are statistically close to random bits sampled from some ideal uniformly random distribution. More precisely, the statistical distance between the output of Pulsar RNG and a uniformly random distribution is bounded by a suitable ε.

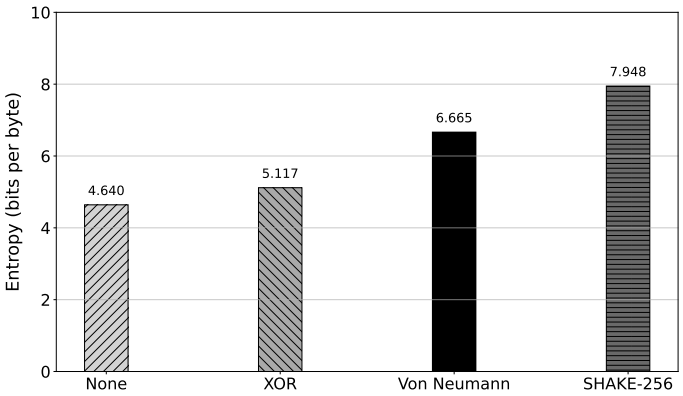


Fig. 4: Entropy for different randomness extractors on data from PSR J0030+0451 (EPTA).

Pulsar	Dataset	Min Entropy (bits per bit)
PSR J0030+0451	EPTA	0.974
PSR J1918-0642	EPTA	0.826
PSR J2124-3358	EPTA	0.801
PSR J1843-1113	EPTA	0.911
PSR J2322+2057	EPTA	0.699
PSR J1832-0836	NANOGrav	0.679
PSR J2302+4442	NANOGrav	0.909
PSR J0030+0451	NANOGrav	0.882
PSR J1918-0642	NANOGrav	0.739
PSR J1012+5307	NANOGrav	0.798

Table 1: Empirical results validating non-trivial (significantly larger than 0) min-entropy for 10 pulsars, 5 from NANOGrav and 5 from EPTA. Note that the maximum possible min-entropy is 1.

3.2 Statistical Tests

Previous analyses of RNGs derived from astrophysical sources rely on black-box statistical tests such as the NIST SP800-22b test [24], `ent` [17], `diehard`, and `dieharder` [25]. In Section 3.2, we show that our Pulsar RNG performs well when evaluated using such statistical tests and provide a discussion regarding debiasing and mixing methods. We note, however, that presenting results for these black-box statistical tests as sole evidence for the suitability of cryptographic RNGs is inaccurate [11]. Even weak (insecure) PRNGs can pass these tests [11]. Therefore, we recommend using our statistical test results only as complementary evidence to our theoretical claims. We show NIST SP800-22b results for the complete version of our Pulsar RNG, including SHA-512 quantification and SHAKE-256 randomness extraction, on the NIST Statistical Testing

NIST test	Proportion	P-value	Pass
Frequency	10/10	0.911413	Y
BlockFrequency	10/10	0.911413	Y
CumulativeSums	10/10	0.534146	Y
Runs	10/10	0.213309	Y
LongestRun	10/10	0.534146	Y
Rank	10/10	0.534146	Y
FFT	10/10	0.534146	Y
ApproximateEntropy	10/10	0.122325	Y
Serial	10/10	0.017912	Y
LinearComplexity	10/10	0.004301	Y

Table 2: Statistical Test Results for Pulsar RNG on PSR J0030+0451 (EPTA).

suite. We test 1 million generated bits evaluated as 10 bitstreams of 100K bits each. Table 2 shows the results for PST J0030+0451 on EPTA Data.

The Pulsar RNG passes all tests in NIST SP800-22b. NIST SP800-22b compares a given bit stream to the null hypothesis of a uniformly random distribution of binary bits [10]. The frequency test checks the fraction of 0s and 1s in the bit stream. The block frequency test checks the same fraction but for segments or *blocks* of the bit stream. The cumulative sum test checks whether the cumulative sum of the bits in the bit stream follows a random walk. The runs test checks the maximum length of consecutive 0s or 1s. The longest runs of ones test checks the maximum length of consecutive 1s in blocks of the bit stream. The Fast Fourier Transform [26] test, FFT for short, checks if there are any repeating patterns in the bit stream. The Approximate Entropy test checks the frequency of all possible overlapping m-bit patterns across the entire sequence. The Serial test focuses on the frequency of all possible overlapping m-bit patterns in the bit stream. Lastly, the Linear Complexity test focuses on the length of a linear feedback shift register (LFSR) to determine whether or not the sequence is complex enough to be considered random [24].

4 Discussion

We have demonstrated the viability of pulsar timing variations as an entropy source for RNGs. Theoretically, we have proved the existence of pulsar-based strong randomness extractors based on reasonable physical assumptions. Experimentally, we have verified the quality of our Pulsar RNG using various standard statistical tests. When compared to TRNGs based on noise in electronic devices, such as Johnson noise in resistors [27], Pulsar RNGs are immune to local temperature fluctuations and other local environmental factors. Pulsar timing variation data is also publicly available from many sources including the North American Nanohertz Observatory for Gravitational Waves [23], the European Pulsar Timing Array [13], the Chinese Pulsar Timing Array [28], and the Parkes Pulsar Timing Array [29] in Australia. Unlike most Quantum RNGs [30], our Pulsar RNG does not require specialized hardware and uses this publicly available data.

In addition to cryptography, our Pulsar RNG is also suitable for many other applications. RNGs are used in Monte Carlo simulations to generate random variates from the underlying distributions of input variables [31]. This random variate generation process is, in fact, the core of the Monte Carlo simulation. RNGs are also used to implement probabilistic data structures such as Bloom Filters [32], Skip Lists [33], and Sketches [34]. Other uses of RNGs include Machine Learning algorithms [35] and even artwork [36]. We have released a fully open-source Python implementation of our Pulsar RNG. Our implementation contains a usable tool to generate random numbers from pulsar data under multiple configurations. The tool currently supports NANOGrav and EPTA data but our modular implementation makes the tool easy to extend for other public datasets. In addition to our tool, we have also open sources all our data processing scripts, randomness extraction methods, and evaluation code. Lastly, we have also publicly released the raw bitstreams we generated to allow an independent verification of our results. We have made all the discussed artifacts available at github.com/jadidbourbaki/pulsar_rng.

Many open problems emerge from this work. We observe (Table 1) that different pulsars yield different entropy. There are over 3000 known pulsars and a comprehensive study would provide a better understanding of the min-entropy and entropy distributions of pulsar timing variations in generation. The deployment of pullsar-based RNGs in real-work applications will also demonstrate practical advantages or challenges our analysis does not address.

Appendix A Formal Definitions & Proofs

Given set S, we write $x \leftarrow\!\!\$\ S$ to mean that x is sampled uniformly randomly from S. For set S, we denote by $|S|$ the number of elements in S. The same notation is used for a list $\mathcal{L}$. We write variable assignments using $\leftarrow$. If the output is the value of a randomized algorithm, we use $\leftarrow\!\!\$$ instead. For a randomized algorithm A, we write output $\leftarrow \mathrm{A}_r(\mathrm{input}_1, \mathrm{input}_2, \cdots, \mathrm{input}_l)$, where $r \in \mathcal{R}$ are the random coins used by A and $\mathcal{R}$ is the set of possible coins. We consider strings $\{0,1\}^n$ to be elements of the Galois Field $\mathrm{GF}(2^n)$. We shorten random variables to r.v. We assume all adversaries are computationally bound. More precisely, we assume adversaries are restricted to non-uniform probabilistic polynomial time [2].

Definition 1 (Statistical Distance Δ)**.** *Let X, Y be r.v.s with range U.*

$$\Delta(X,Y) = \frac{1}{2}\Sigma_{u \in U}|P[X=u] - P[Y=u]|$$

.

Definition 2 (ε-close)**.** *Let X, Y be r.v.s with range U.*

$$X \approx_\varepsilon Y \equiv \Delta(X,Y) \leq \varepsilon$$

Definition 3 (Min-entropy). *Let X be an r.v. with range U.*

$$H_\infty(X) = -\log_2(\max_{u \in U} P[X = u])$$

Definition 4 (k-source). *R.v X is a k-source if $H_\infty(X) \geq k$*

We base our analysis on the following assumption regarding Pulsar timing variations.

Assumption 1. *Let P_X be an r.v. representing timing variation in pulsar signals for pulsar P with universe U. We assume P_X is a k-source (Definition 4) with non-trivial k.*

We can now precisely define a randomness extractor [37] in the cryptographic sense.

Definition 5 (Randomness-Extractor). *Let seed U_d be uniformly distributed on $\{0,1\}^d$. $\mathcal{E} : \{0,1\}^n \times \{0,1\}^d \mapsto \{0,1\}^m$ is a (k, ε)-extractor if, for all k-sources X on $\{0,1\}^n$ independent of U_d,*

$$\mathcal{E}(X, U_d), U_d) \approx_\varepsilon (U_m, U_d)$$

where U_m is uniformly distributed on $\{0,1\}^m$ independent of X and U_d.

Extractors, as defined above, are also referred to in the literature as **strong** extractors.

Definition 6 (Universal hash family). *A family $\mathcal{H}$ of hash functions of size 2^d from $\{0,1\}^n$ to $\{0,1\}^m$ is called universal if, for every $x, y \in \{0,1\}^n$ with $x \neq y$,*

$$P_{h \in \mathcal{H}}[h(x) = h(y)] \leq 2^{-m}.$$

We denote our Pulsar RNG algorithm as $\mathcal{E}_p$. $\mathcal{E}_p$ relies on a universal hash family. $\mathcal{E}_p$ takes quantified data from a Pulsar entropy source $x_p \leftarrow\!\!\$\ P_X$. It then uses a hash function from a universal hash family $h_p \leftarrow\!\!\$\ \mathcal{H}$ of size 2^d. In our default implementation, this is the SHAKE-256 hash function from the SHA-3 family of hashes. $\mathcal{E}_p$ then uses p_x as the seed for h_p.

$$\mathcal{E}_p(p_x, h) = h_p(p_x)$$

There is a well-known result in cryptography called the Leftover Hash Lemma [37], originally proved by [38]. The Leftover Hash Lemma proves that a universal hash family can be used to construct a strong extractor from a k-source.

Theorem 1 (Leftover hash lemma). *Let X be a k-source with universe U. Fix $\varepsilon > 0$. Let $\mathcal{H}$ be a universal hash family of size 2^d with output length $m = k - 2\log_2(\frac{1}{\varepsilon})$. Define*

$$\mathcal{E}(x, h) = h(x)$$

Then $\mathcal{E}$ is a strong $(k, \varepsilon/2)$ extractor with seed length d and output length m.

We are now ready to prove our main result, that our Pulsar RNG $\mathcal{E}_p$ is a strong extractor.

Theorem 2. *Let P_X be an r.v. representing timing variation in pulsar signals for pulsar P with universe U. Fix $\varepsilon > 0$. Pulsar RNG, $\mathcal{E}_p$ is a strong $(m + 2\log_2(\frac{1}{\varepsilon}))$-extractor with seed length d and output length m.*

Proof The proof follows directly from Assumption 1 and the Leftover Hash Lemma. □

References

[1] Katz, J., Lindell, Y.: Introduction to Modern Cryptography, Second Edition, 2nd edn. Chapman & Hall/CRC, n/a (2014)

[2] Pass, R., Shelat, A.: A Course in Cryptography. Lecture Notes. Available at: https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf (2010)

[3] Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978) https://doi.org/10.1145/359340.359342

[4] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology, pp. 10–18. Springer, Berlin, Heidelberg (1985)

[5] Jun, B., Kocher, P.: The intel random number generator. Cryptography Research Inc. white paper **27**(1-8), 66 (1999)

[6] Pimbblet, K.A., Bulmer, M.: Random numbers from astronomical imaging. Publications of the Astronomical Society of Australia **22**(1), 1–5 (2005) https://doi.org/10.1071/AS04043

[7] Chapman, E., Grewar, J., Natusch, T.: Celestial sources for random number generation. Australian Information Security Management Conference (2016)

[8] Lee, J.S., Cleaver, G.B.: The cosmic microwave background radiation power spectrum as a random bit generator for symmetric- and asymmetric-key cryptography. Heliyon **3**(10), 00422 (2017) https://doi.org/10.1016/j.heliyon.2017.e00422

[9] Wu, C., Bai, B., Liu, Y., Zhang, X., Yang, M., Cao, Y., Wang, J., Zhang, S., Zhou, H., Shi, X., Ma, X., Ren, J.-G., Zhang, J., Peng, C.-Z., Fan, J., Zhang, Q., Pan, J.-W.: Random number generation with cosmic photons. Phys. Rev. Lett. **118**, 140402 (2017) https://doi.org/10.1103/PhysRevLett.118.140402

[10] Dawson, J.R., Hobbs, G., Gao, Y., Camtepe, S., Pieprzyk, J., Feng, Y., Tranfa, L., Bradbury, S., Zhu, W., Li, D.: Physical publicly verifiable randomness from pulsars. Astronomy and Computing **38**, 100549 (2022) https://doi.org/10.1016/j.ascom.2022.100549

[11] Saarinen, M.-J.O.: SP 800–22 and GM/T 0005–2012 Tests: Clearly Obsolete, Possibly Harmful . In: 2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 31–37. IEEE Computer Society, Los Alamitos, CA, USA (2022). https://doi.org/10.1109/EuroSPW55150.2022.00011 . https://doi.ieeecomputersociety.org/10.1109/EuroSPW55150.2022.00011

[12] Matthews, A.M., Nice, D.J., Fonseca, E., Arzoumanian, Z., Crowter, K., Demorest, P.B., Dolch, T., Ellis, J.A., Ferdman, R.D., Gonzalez, M.E., *et al.*: The nanograv nine-year data set: astrometric measurements of 37 millisecond pulsars. The Astrophysical Journal **818**(1), 92 (2016)

[13] EPTA, InPTA, *et al.*: The second data release from the european pulsar timing array: V. search for continuous gravitational wave signals. Astronomy and Astrophysics **690**, 118 (2024)

[14] Luo, J., Ransom, S., Demorest, P., Ray, P.S., Archibald, A., Kerr, M., Jennings, R.J., Bachetti, M., van Haasteren, R., Champagne, C.A., Colen, J., Phillips, C., Zimmerman, J., Stovall, K., Lam, M.T., Jenet, F.A.: PINT: A Modern Software Package for Pulsar Timing. The Astrophysical Journal **911**(1), 45 (2021) https://doi.org/10.3847/1538-4357/abe62f arXiv:2012.00074 [astro-ph.IM]

[15] Doran, R.W.: The gray code. Technical report, Citeseer (2007)

[16] Penard, W., Van Werkhoven, T.: On the secure hash algorithm family. Cryptography in context, 1–18 (2008)

[17] Walker, J.: ENT: A Pseudorandom Number Sequence Test Program. Fourmilab: Switzerland, 2008 (2008)

[18] Walker, J.: Hotbits: Genuine random numbers, generated by radioactive decay. Online: http://www. fourmilab. ch/hotbits (2001)

[19] Stipčević, M., Koç, Ç.K.: True random number generators. In: Open Problems in Mathematics and Computational Science, pp. 275–315. Springer, ??? (2014)

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

[21] Antonelli, M., Basu, A., Haskell, B.: Stochastic processes for pulsar timing noise: fluctuations in the internal and external torques. Monthly Notices of the Royal Astronomical Society **520**(2), 2813–2828 (2023) https://doi.org/10.1093/mnras/stad256

[22] Zubieta, E., García, F., del Palacio, S., Araujo Furlan, S. B., Gancio, G., Lousto, C. O., Combi, J. A., Espinoza, C. M.: Timing irregularities and glitches from the pulsar monitoring campaign at iar. A&A **689**, 191 (2024) https://doi.org/10.1051/0004-6361/202450441

[23] Agazie, G., Anumarlapudi, A., Archibald, A.M., Arzoumanian, Z., Baker, P.T., Bécsy, B., Blecha, L., Brazier, A., Brook, P.R., Burke-Spolaor, S., Burnette, R., Case, R., Charisi, M., Chatterjee, S., Chatziioannou, K., Cheeseboro, B.D., Chen, S., Cohen, T., Cordes, J.M., Cornish, N.J., Crawford, F., Cromartie, H.T., Crowter, K., Cutler, C.J., DeCesar, M.E., DeGan, D., Demorest, P.B., Deng, H., Dolch, T., Drachler, B., Ellis, J.A., Ferrara, E.C., Fiore, W., Fonseca, E., Freedman, G.E., Garver-Daniels, N., Gentile, P.A., Gersbach, K.A., Glaser, J., Good, D.C., Gültekin, K., Hazboun, J.S., Hourihane, S., Islo, K., Jennings, R.J., Johnson, A.D., Jones, M.L., Kaiser, A.R., Kaplan, D.L., Kelley, L.Z., Kerr, M., Key, J.S., Klein, T.C., Laal, N., Lam, M.T., Lamb, W.G., W. Lazio, T.J., Lewandowska, N., Littenberg, T.B., Liu, T., Lommen, A., Lorimer, D.R., Luo, J., Lynch, R.S., Ma, C.-P., Madison, D.R., Mattson, M.A., McEwen, A., McKee, J.W., McLaughlin, M.A., McMann, N., Meyers, B.W., Meyers, P.M., Mingarelli, C.M.F., Mitridate, A., Natarajan, P., Ng, C., Nice, D.J., Ocker, S.K., Olum, K.D., Pennucci, T.T., Perera, B.B.P., Petrov, P., Pol, N.S., Radovan, H.A., Ransom, S.M., Ray, P.S., Romano, J.D., Sardesai, S.C., Schmiedekamp, A., Schmiedekamp, C., Schmitz, K., Schult, L., Shapiro-Albert, B.J., Siemens, X., Simon, J., Siwek, M.S., Stairs, I.H., Stinebring, D.R., Stovall, K., Sun, J.P., Susobhanan, A., Swiggum, J.K., Taylor, J., Taylor, S.R., Turner, J.E., Unal, C., Vallisneri, M., Haasteren, R., Vigeland, S.J., Wahl, H.M., Wang, Q., Witt, C.A., Young, O., Collaboration, T.N.: The nanograv 15 yr data set: Evidence for a gravitational-wave background. The Astrophysical Journal Letters **951**(1), 8 (2023) https://doi.org/10.3847/2041-8213/acdac6

[24] Bassham III, L.E., Rukhin, A.L., Soto, J., Nechvatal, J.R., Smid, M.E., Barker, E.B., Leigh, S.D., Levenson, M., Vangel, M., Banks, D.L., et al.: Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards & Technology (2010)

[25] Brown, R.G., Eddelbuettel, D., Bauer, D.: Dieharder. Duke University Physics Department Durham, NC, 27708–0305 (2018)

[26] Heideman, M., Johnson, D., Burrus, C.: Gauss and the history of the fast fourier transform. IEEE Assp Magazine **1**(4), 14–21 (1984)

[27] Tyson, T.: Thermal Johnson Noise Generated by a Resistor. https://123.physics.ucdavis.edu/week_2_files/Johnson_noise_intro.pdf (2013)

[28] Xu, H., Chen, S., Guo, Y., Jiang, J., Wang, B., Xu, J., Xue, Z., Nicolas Caballero, R., Yuan, J., Xu, Y., Wang, J., Hao, L., Luo, J., Lee, K., Han, J., Jiang, P., Shen, Z., Wang, M., Wang, N., Xu, R., Wu, X., Manchester, R., Qian, L., Guan, X., Huang, M., Sun, C., Zhu, Y.: Searching for the nano-hertz stochastic gravitational wave background with the chinese pulsar timing array data release i. Research in Astronomy and Astrophysics **23**(7), 075024 (2023) https://doi.org/10.1088/1674-4527/acdfa5

[29] Manchester, R.N., Hobbs, G., Bailes, M., Coles, W.A., van Straten, W., Keith,

M.J., Shannon, R.M., Bhat, N.D.R., Brown, A., Burke-Spolaor, S.G., Champion, D.J., Chaudhary, A., Edwards, R.T., Hampson, G., Hotan, A.W., Jameson, A., Jenet, F.A., Kesteven, M.J., Khoo, J., Kocz, J., Maciesiak, K., Oslowski, S., Ravi, V., Reynolds, J.R., Sarkissian, J.M., Verbiest, J.P.W., Wen, Z.L., Wilson, W.E., Yardley, D., Yan, W.M., You, X.P.: The Parkes Pulsar Timing Array Project. Publications of the Astronomical Society of Australia **30**, 017 (2013) https://doi.org/10.1017/pasa.2012.017 arXiv:1210.6130 [astro-ph.IM]

[30] Ma, X., Yuan, X., Cao, Z., Qi, B., Zhang, Z.: Quantum random number generation. npj Quantum Information **2**(1), 1–9 (2016)

[31] Raychaudhuri, S.: Introduction to monte carlo simulation. In: 2008 Winter Simulation Conference, pp. 91–100 (2008). IEEE

[32] Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM **13**(7), 422–426 (1970) https://doi.org/10.1145/362686.362692

[33] Pugh, W.: Skip lists: a probabilistic alternative to balanced trees. Commun. ACM **33**(6), 668–676 (1990) https://doi.org/10.1145/78973.78977

[34] Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms **55**(1), 58–75 (2005) https://doi.org/10.1016/j.jalgor.2003.12.001

[35] Mitchell, T.: Introduction to machine learning. Machine learning **7**, 2–5 (1997)

[36] Bauer, A.: Gallery of random art. https://www.random-art.org/ (1998)

[37] Reyzin, L.: Lecture Notes for CS 937: Advanced Topics in Cryptography. Spring 2011. Available online: https://www.cs.bu.edu/~reyzin/teaching/s11cs937/notes-leo-1.pdf (2011)

[38] Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing. STOC '89, pp. 12–24. Association for Computing Machinery, New York, NY, USA (1989). https://doi.org/10.1145/73007.73009 . https://doi.org/10.1145/73007.73009