proceedings

Proceedings

Auditable Blockchain Randomization Tool [†]

Olivia Saa [‡] and Julio Michael Stern [*,‡]

IME-USP—Institute of Mathematics and Statistics of the University of São Paulo, Rua do Matão 1010, 05508-090 São Paulo, Brazil; olivia@ime.usp.br or olivia.saa@iota.org

* Correspondence: jstern@ime.usp.br

[†] Presented at the 39th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Garching, Germany, 30 June–5 July 2019.

[‡] These authors contributed equally to this work.

Published: 2 December 2019

Abstract: Randomization is an integral part of well-designed statistical trials, and is also a required procedure in legal systems. Implementation of honest, unbiased, understandable, secure, traceable, auditable and collusion resistant randomization procedures is a mater of great legal, social and political importance. Given the juridical and social importance of randomization, it is important to develop procedures in full compliance with the following desiderata: (a) Statistical soundness and computational efficiency; (b) Procedural, cryptographical and computational security; (c) Complete auditability and traceability; (d) Any attempt by participating parties or coalitions to spuriously influence the procedure should be either unsuccessful or be detected; (e) Open-source programming; (f) Multiple hardware platform and operating system implementation; (g) User friendliness and transparency; (h) Flexibility and adaptability for the needs and requirements of multiple application areas (like, for example, clinical trials, selection of jury or judges in legal proceedings, and draft lotteries). This paper presents a simple and easy to implement randomization protocol that assures, in a formal mathematical setting, full compliance to the aforementioned desiderata for randomization procedures.

Keywords: blockchain entropy; statistical randomization; judicial sortition

> *Meos tam suspicione quam crimine judico carere oportere.*
> My people should be free from either crime or suspicion.
> Julius Caesar (62BC), in Suetonius (119CE, Sec.I.74.2).

1. Introduction: Bad and Good Practices in Randomization

Randomization is a technique used in the design of statistical experiments: in a clinical trial, for example, patients are randomly assigned to distinct groups receiving different treatments with the goal of studding and contrasting their effects. Randomization is nowadays considered a golden standard in statistical practice; its motivation is to prevent systematic biases (like an unfair or tendentious assignment process) that could distort (unintentionally or purposely) the conclusions of the study. For further comments on randomization see [1–3], for Bayesian perspectives see [4,5]. In the legal context, randomization (also known as sortition or allotment) is routinely used for the selection of jurors or judges assigned to a given judicial case; see [6]. For these applications, our initial quotation, from the Roman emperor Julius Caesar, suggests the highest standards of technical quality, and auditability, see [7].

Rerandomization is the practice of rejecting and discarding (for whatever reason) a given randomized outcome, that is subsequently replaced by a new randomization. Repeated rerandomization can be used to completely circumvent the haphazard, unpredictable or aimless nature of randomization, allowing a premeditated selection of a final outcome of choice. There are advanced statistical techniques capable of blending the best characteristics of random and intentional sampling, see for example [8–12]. Nevertheless, rerandomization is often naively used, or abused, with the excuse of (subjectively) "avoiding outcomes that do not look random enough", see for example [13,14]. In the legal context, spurious manipulations of the randomization process are often linked to fraud, corruption and similar maladies, see [6] and references therein.

In order to comply with the best practices for randomization processes, the authors of [6] recommend the use of computer software having a long list of characteristics, for example, being efficient and fully auditable, well-defined and understandable, sound and flexible, secure and transparent. Such requirements are expressed by the following (revised) *desiderata* for randomization procedures:

> *Given the juridical and social importance of the themata under scrutiny, we believe that it is important to develop randomization procedures in full compliance with the following desiderata: (a) Statistical soundness and computational efficiency, see [15–18]; (b) Procedural, cryptographical and computational security, see [19–22]; (c) Complete auditability and traceability, see [23–25]; (d) Any attempt by participating parties or coalitions to spuriously influence the procedure should be either unsuccessful or be detected, see [26–28]; (e) Open-source programming; (f) Multiple hardware platform and operating system implementation; (g) User friendliness and transparency, see [29,30]; (h) Flexibility and adaptability for the needs and requirements of multiple application areas (like, for example, clinical trials, selection of jury or judges in legal proceedings, and draft lotteries), see [6].*

Such requirements conflate several complementary characteristics that may seem, at first glance, incompatible. For example, strong security is often (but wrongly) associated with excessive secrecy, a doctrine known as "security by obscurity", computer routines may be efficient but are often tough as hard to audit, and mathematically well-defined algorithms may be perceived as hard to understand. The bibliographical references given in the formerly stated *desiderata for randomization procedures* already hint at technologies that can be used to achieve a fully compliant randomization procedure, most preeminently, the blockchain. This is the key technology supporting modern public ledgers, cryptocurencies, and a host of related applications.

A technical challenge for the application under scrutiny is the generation of pseudo-random number sequences that reconcile complementary properties related to computational efficiency, statistical soundness, and cryptographic security. In this respect, the excellent statistical and computational characteristics of linear recurrence pseudo-random number generators (or their modern descendants and relatives), like [16], can be reconciled with the needs concerning unpredictability and cryptographic security by appropriate starts and restarts of the linear recurrence generator. A sequence start for a linear recurrence generator is defined by a *seed* specified by a vector of (typically 1 to 64) integers, while a restart is defined by a *jump-ahead* or *skip-ahead* specified by a single integer (kept small relative to the generator's full period), see [22].

Unpredictable and cryptographically secure *seeds* and *jump-aheads* can be provided by high entropy bit streams extracted from blockchain transactions, an idea that has already been explored in the works of [31–34].

The next section develops a possible implementation of a fully compliant core randomization protocol based on blockchain technology, and also makes a simple prototype available for study and further research. Moreover, in order to make it simple and easy to use, we develop the prototype on top of a readily available crypto-currency platform. We use Bitcoin for this example, but other alternatives like Ethereum or other cryptocurrencies whose miners work under the same incentives model can be used with minor adaptations.

2. Results: Core Randomization Protocol in Blockchain

We intend to establish a protocol able to deliver on demand pseudo random numbers, from an auditable and immutable ledger. The procedure will start as follows: the user (the part that wants to receive a random number) shall send a Bitcoin transaction with a register of its purpose embedded in it. (One way to embed a message in a transaction is using the OP_RETURN script, which allows to store up to 40 bytes in a transaction.) The recipient of this transaction may be a proxy representing a competent authority, a pertinent regulatory agency, an agreed custodian, etc. When this transaction is first attached to the blockchain, we concatenate the transaction ID (a 32 bytes, hexadecimal number) and the block header (a 80 bytes, hexadecimal number). In case someone tries to generate more than one transaction for a same purpose, just take the one that was attached first. The resulting 112 bytes hexadecimal number will be the input for some known Verifiable Delay Function (VDF), that should be calibrated accordingly to the purpose of the random number. For instance, a less critical purpose should have a VDF that delays the result in just a few seconds, or even skip completely the VDF step. A critical purpose, with significant interests involved, should have a more complex VDF, with a delay of minutes or even hours. The final result, after the VDF, will be the source for our seeds and jump-aheads.

With the aid of this protocol, one is able to find a different pseudo-random number for each user that demands it. Note that the user does not have any incentive to try to modify its transaction ID, because he does not have any control of the block header. We assume that the user and the miner are not the same person, so a miner will only be interested in trying to control his block header if he is paid to do so. Since the last stage of our protocol involves the calculation of a VDF, it will take a certain amount of time to the miner to decide if the the block he has found will be of interest of the user. Thus, he might even lose his block, if some other miner broadcasts a block of his own before he finishes calculating the VDF.

In the following subsection, the miner's payoff and the necessary delay T for the Verifiable Delay Functions will be explicitly calculated.

2.0.1. Preventing Collusion for Spurious Manipulation

Suppose a malicious user tries to bribe a miner that controls a fraction p of the network's computational power. A prize $P = nB$, where B is the Bitcoin block reward, will be paid to the miner if he successfully mines what we call a "desirable block": a block that will deliver a random number in a set A, chosen by the malicious user. Let also λ be the average rate of incoming blocks and q the probability of a randomly generated number being an element of A, i.e., the measure of the set of desirable results for the malicious user. Finally, let T be the expected amount of time needed for the VDF calculations. The moment a miner finds a block that can be accepted by the network, he faces the decision of broadcasting it before checking the VDF, or calculating the VDF before broadcasting. If he decides to check the VDF before broadcasting, he might start another attempt to find a block rightaway.

First, we calculate the expected absolute payoff for the first and second options, called $\mathbb{E}_1$ and $\mathbb{E}_2$, respectively. $\mathbb{E}_1$ will be larger than B, since the miner might issue a desirable block by chance:

$$\mathbb{E}_1 = B + qP = B(1 + nq) \tag{1}$$

On the other hand, if the miner chooses to calculate the VDF, he will receive the block reward and the prize P, but with a probability given by

$$\begin{aligned}\mathbb{E}_2 =&(B+P)q\mathbb{P}\{\text{no other node finding a block before } t=T\} \\ &+(B+P)(1-q)\mathbb{P}\{\text{successfully mining a desirable block in another attempt}\} \\ =&B(1+n)q\exp(-(1-p)\lambda T) \\ &+B(1+n)(1-q)\sum_{i=1}^{\infty}\mathbb{P}\{\text{successfully mining a desirable block after } i \text{ attempts}\}\end{aligned} \tag{2}$$

The probabilities inside the summation, in the last equation, can be calculated as the product of the probability of finding a desirable block after i attempts (that will be a geometric distribution with probability of success q) and the probability of finding and checking i blocks before the rest of the network mines one.

Considering

$$\begin{aligned}&P\{\text{attacker finding and analyzing } i \text{ blocks before another node mining one}\} \\ &=\int_{t=0}^{\infty} p\lambda\exp(-p\lambda t)\frac{(p\lambda t)^{i-1}}{(i-1)!}\exp(-(1-p)\lambda(t+T))dt \\ &=\frac{(p\lambda)^i\exp(-(1-p)\lambda T)}{(i-1)!}\int_{t=0}^{\infty}\exp(-\lambda t)t^{i-1} \\ &=\frac{(p\lambda)^n\exp(-(1-p)\lambda T)}{(i-1)!}\lambda^{-i}(i-1)! \\ &=p^i\exp(-(1-p)\lambda T)\end{aligned}$$

it follows that

$$\begin{aligned}\mathbb{E}_2 =&B(1+n)\left[q\exp(-(1-p)\lambda T)+(1-q)\sum_{i=1}^{\infty}q(1-q)^{i-1}p^i\exp(-(1-p)\lambda T)\right] \\ =&B(1+n)\exp(-(1-p)\lambda T)\left(q+\frac{(1-q)pq}{1-p+pq}\right)\end{aligned} \tag{3}$$

Finally, in order to make accepting the bribe not lucrative, we must have $\mathbb{E}_1 > \mathbb{E}_2$, i.e.:

$$\lambda T > \frac{1}{1-p}\log\left(\frac{1+n}{1+nq}\frac{q}{1-p+pq}\right) \tag{4}$$

Since for every $n>0$ we have $\frac{1+n}{1+nq} < \frac{1}{q}$, if we choose $\lambda T^* = \frac{1}{1-p}\log\left(\frac{1}{q}\frac{q}{1-p+pq}\right)$, we guarantee that the attack will not be lucrative for any bribe $P = nB$. Also, since it can be assumed that $p < 1/2$, a value $\lambda T^* = 2\log\left(\frac{2}{1+q}\right) < 2\log(2)$ will be high enough to prevent an attack for any bribe and any acceptable value of p.

3. Conclusions and Final Remarks

We formalized a simple and effective protocol to generate on demand pseudo random numbers, in a fully auditable way. We have demonstrated that none of the involved parts has enough financial incentives to try to affect the random number outcome: the part that issues the transaction lacks this power, since it does not have any control on the block header; and the miners do not have enough financial incentives to collude with an attacker, provided a suitable Verifiable Delay Function is applied.

The essentially decentralized, yet completely traceable and auditable nature of the protocol presented in this article, makes the resulting randomization process eminently reliable without recourse of blind trust in any central authority. The authors believe the adoption of such a protocol by the the Brazilian Supreme Court (STF), as recommended in [6], would significantly increase public confidence in the judicial system and be a contributing factor for political and social stability. A simple prototype of the randomization tool described in this article is available in the supplementary materials; it is not intended to be used in a full-fledged application, but only to provide a working example of the key procedures.

Supplementary Materials: A simple prototype of the randomization tool described in this article is available online at https://github.com/oliviasaa/random_generator/blob/master/random_generator.py.

Funding: This research was funded by FAPESP – the State of São Paulo Research Foundation (grants CEPID-CeMEAI 2013/07375-0 and CEPID-Shell-RCGI 2014/50279-4) and CNPq – the Brazilian National Counsel of Technological and Scientific Development (grant PQ 301206/2011-2, 307648/2018-4 and GD 140490/2016-7).

Acknowledgments: The authors are grateful for the support received from IME-USP—the Institute of Mathematics and Statistics of the University of São Paulo and for the advice of Prof. Serguei Popov. The authors also received support from ABJ—the Brazilian Jurimetrics Association; STF—Supremo Tribunal Federal (the Brazilian Supreme Court), which motivated the study and provided the data analysed in Marcondes et al. [6]; the IOTA Foundation; and received helpful comments and advice from Adilson Simonis, Álvaro Machado Dias, Julio Trecenti, Rafael Bassi Stern and Marcelo Guedes Nunes.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

STF	Superior Tribunal Federal—Brazilian Supreme Court
VDF	Verifiable Delay Function

References

1. Pearl, J. *Causality: Models, Reasoning, and Inference*; Cambridge University Press: Cambridge, UK, 2000.
2. Pearl, J. *Simpson's Paradox: An Anatomy*; Tech. Rep.; University of California: Los Angeles, CA, USA, 1983.
3. Stern, J.M. Decoupling, Sparsity, Randomization, and Objective Bayesian Inference. *Cybern. Hum. Know.* **2008**, *15*, 49–68.
4. Basu, D.; Ghosh, J.K. *Statistical Information and Likelihood, A Collection of Essays by Dr.Debabrata Basu*; Springer: Berlin, Germany, 1988.
5. Gelman, A.; Carlin, J.B.; Stern, H.S.; Rubin, D.B. *Bayesian Data Analysis*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2003.
6. Marcondes, D.; Peixoto, C.; Stern, J.M. Assessing Randomness in Case Assignment: The Case Study of the Brazilian Supreme Court. *Law Probab. Risk* **2019**, doi:10.1093/lpr/mgz006.
7. Tranquillus, S. *The Lives of the Caesars*; Harvard University Press: Cambridge, MA, USA, 1979; Volume 1.
8. Fossaluza, V.; Lauretto, M.S.; Pereira, C.A.B.; Stern, J.M. Combining Optimization and Randomization Approaches for the Design of Clinical Trials. *Springer Proc. Math. Stat.* **2015**, *118*, 173–184.
9. Lauretto, M.S.; Nakano, F.; Pereira, C.A.B.; Stern, J.M. Intentional Sampling by Goal Optimization with Decoupling by Stochastic Perturbation. *Am. Inst. Phys. Conf. Proc.* **2012**, *1490*, 189–201.
10. Lauretto, M.S.; Stern, R.B.; Morgan, K.L.; Clark, M.H.; Stern, J.M. Haphazard Intentional Sampling and Censored Random Sampling to Improve Covariate Balance in Experiments. *Am. Inst. Phys. Conf. Proc.* **2017**, doi:10.1063/1.4985356.

11. Morgan, K.L.; Rubin, D.B. Rerandomization to improve covariate balance in experiments. *Ann. Stat.* **2012**, *40*, 1263–1282.
12. Morgan, K.L.; Rubin, D.B. Rerandomization to balance tiers of covariates. *J. Am. Assoc.* **2015**, *110*, 1412–1421.
13. Bruhn, M.; McKenzie, D. In Pursuit of Balance: Randomization in Practice in Development Field Experiments. *Am. Econ. J. Appl. Econ.* **2009**, *1* 200–232.
14. Ruxton, G.D.; Colegrave, N. *Experimental Design for the Life Sciences*, 2nd ed.; Oxford University Press: Oxford, UK, 2006.
15. Hammersley, J.M.; Handscomb, D.C. *Monte Carlo Methods*; Chapman and Hall: London, UK, 1964.
16. Haramoto, H.; Matsumoto, M.; Nishimura, T.; Panneton, F.; L'Ecuyer, P. Efficient Jump Ahead for F2-Linear Random Number Generators. *INFORMS J. Comput.* **2008**, *20*, 290–298.
17. Knuth, D.E. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed.; Addison-Wesley Longman Publ. Co., Inc.: Reading, MA, USA, 1997.
18. Ripley, B.D. *Stochastic Simulation*; Wiley: Hoboken, NJ, USA, 1987.
19. Aumasson, J.-P. *Serious Cryptography: A Practical Introduction to Modern Encryption*; No Starch Press: San Francisco, CA, USA, 2017.
20. Boyar, J. Inferring Sequences Produced by Pseudo-Random Number Generators. *J. ACM* **1989**, *36*, 129–141.
21. Katz, J.; Lindell, Y. *Introduction to Modern Cryptography*; Chapman and Hall: London, UK, 2014.
22. L'Ecuyer, P. Random number generation. In *Handbook of Computational Statistics*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 35–71.
23. Haber, S.; Stornetta, W. How to time-stamp a digital document. *J. Cryptol.* **1991**, *3*, 99–111.
24. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Unaffiliated Technical Report. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 30 June 2019).
25. Wattenhofer, R. *Distributed Ledger Technology: The Science of Blockchain*; Inverted Forest: Scotts Valley, CA, USA, 2017.
26. Boneh, D.; Bonneau, J.; Bünz, B.; Fisch, B. Verifiable Delay Functions. Cryptology ePrint Archive, Report 2018/601. 2018. Available online: https://eprint.iacr.org/2018/601 (accessed on 30 June 2019).
27. Goldschlag, D.M.; Stubblebine, S.G. Publicly Verifiable Lotteries: Applications of Delaying Functions. In *International Conference on Financial Cryptography*; Springer: Berlin, Germany, 1998; pp. 214–226.
28. Rabin, M.O. Transaction protection by beacons. *J. Comput. Syst. Sci.* **1983**, *27*, 256–267.
29. Parikh, R.; Pauly, M. What Is Social Software? In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 1973; pp. 3–13.
30. Stern, J.M. Verstehen (causal/interpretative understanding), Erklären (law-governed description/prediction), and Empirical Legal Studies. *J. Inst. Theor. Econ.* **2018**, *174*, 105–114.
31. Bonneau, J.; Clark, J.; Goldfeder, S. On Bitcoin as a public randomness source. *IACR Cryptol. ePrint Arch.* **2015**, *2015*, 1015.
32. Kelsey, J.; Schneier, B.; Hall, C.; Wagner, D. Secure applications of low-entropy keys. In *International Workshop on Information Security*; Springer: Berlin, Germany, 1997.
33. Pierrot, C.; Wesolowski, B. Malleability of the Blockchain's Entropy. 2016. Available online: https://eprint.iacr.org/2016/370.pdf (accessed on 30 June 2019).
34. Popov, S. On a decentralized trustless pseudo-random number generation algorithm. *J. Math. Cryptol.* **2017**, *11*, 37–43.