Random numbers generation: tests and attacks

Sylvain Guilley[1,2,3] and Youssef El Housni[1]

[1] Secure-IC S.A.S., 15 Rue Claude Chappe, Bât. B, 35 510 Cesson-Sévigné, FRANCE
[2] LTCI, Télécom ParisTech, Université Paris-Saclay, 75 013 Paris, FRANCE
[3] École Normale Supérieure, 45 Rue d'Ulm, 75 005 Paris, FRANCE

***Abstract*—The generation of random numbers is a keystone function in any cryptographic protocol. Indeed, in a security context, the random numbers generation shall withstand assaults from adversaries. It is thus paramount to validate both its *functionality* and its *robustness* in front of attacks, including *fault injection attacks*. The verification implies tests, which shall thus be carried out in nominal but also in perturbed operational environments.**

In this paper, we review standard tests already existing and still under development. As a first contribution, we suggest a new kind of metrics to assess the quality of the random sequences of bits. As a second contribution, we analyse fault injections in true random number generators and explore whether such faulted behavior can be self-induced within the circuit itself. This analysis reveals a plausible interpretation of the behavior of circuits based on the analysis of long term noise, e.g., TRNGs based on ring oscillators.

Key words: true random number generators (TRNGs), cryptographic applications, randomness test, resilience to attacks, physically unclonable functions (PUFs).

I. INTRODUCTION

It is well known that security applications rely on random elements, such as:

- **keys**: in protocols based on symmetric & asymmetric cryptography, and applications such as authentication;
- **initialization vectors (IVs)**: in modes of operations, where it is required that IVs are explicit;
- **nonces**: to make each signature unique;
- **noise**: as required in code- and lattice-based (post-quantum) cryptography [16];
- etc.

Those random numbers or elements shall not be guessable by an attacker. The requirement is fairly simple to express: those n-bit elements shall spawn from an ideal distribution, namely $\mathcal{B}(n, 1/2)$ (binomial law with probability $p = 1/2$ for each of the n bits of the considered element). However, the question is: "how to assess that the source distribution is indeed binomially distributed?" Clearly, mathematical generation only produces pseudo-random numbers, because the n bits are not independent, hence are not distributed as prescribed. The *real* sources of entropy must thus be *physical*. It has been validated (for instance by the validation of the violation of Bell inequalities) that the world is quantum: particular behavior cannot be described by hidden variables (such as the seed of a pseudo-random number generation function). In practice, many physical noise sources are of quantum origin, such as thermal noise, flicker noise, etc. The key question is now to assess whether the generated random numbers stem from a truly random number generator (TRNG).

The rest of this position paper reviews two important facets of secure random number generation. First, we describe existing and developing standard tests in Sec. II. We contrast them and suggest possible improvements in Sec. III, where we give directions to improve the nature of tests (based on cryptographic properties of Boolean functions). Second, we review in Sec. IV some examples of secure TRNGs which happen to fail in adversarial conditions. We also discuss some situations where a TRNG secure in a mode of operation can fail in different modes. Eventually, we conclude in Sec. V on existing tests and on the challenges to maintain high quality TRNG in adversarially perturbed environments. An appendix A details application of tests for TRNG to another emerging domain (from the standardization standpoint), namely the Physically Unclonable Functions.

II. CURRENT STANDARDS

There exist several standards, detailed hereafter, to assess the quality of the random numbers generated by secure devices. Those specify methods, which can be applied to attest for two features:

- checking the functionality, and
- validating its robustness under attack.

The two-step process is depicted in Fig. 1. This way to proceed is implicitly put forward in standard practices, such as for instance tests (e.g., ISO-19790 [11]) making assumptions on operational environment and then how to validate the tests when the device under test is immersed in its operational (i.e., untrustworthy) environment (e.g., as documented in ISO 20540 [7]).

Existing tests are now discussed in the following subsections.

A. DieHard

The DieHard test suite has been produced over several years by Marsalia [14]. It requires a lot of data, typically 80 million of bits. DieHard is today considered the ancestor of NIST FIPS SP 800 22 (refer to Sec. II-C).

978-1-5386-8197-8/18/$31.00 ©2018 IEEE
DOI 10.1109/FDTC.2018.00016

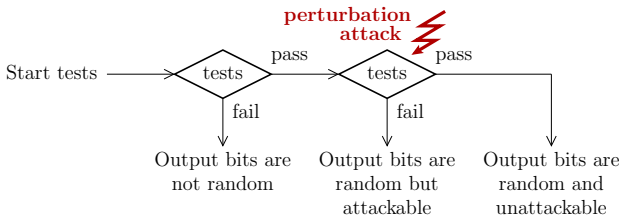

Figure 1. Two-step method to validate functionality and robustness of a TRNG in an adversarial environment, based on randomness tests such as those discussed in Sec. II

B. NIST FIPS 140-2

The random tests in FIPS 140-2 [21] originally consisted in four tests, namely monobit, poker, runs, long runs. The sequence is short (only 20,000 bits).

These tests are today deprecated. See next sections for up-to-date tests. Still, current version of FIPS 140-2 keeps a simple health test: the first 16-bit (or greater) block is compared to the previous first 16-bit block of the next sequence. An alarm is generated if they are the same (which is a possible testimony for a stuck-at issue).

C. NIST FIPS SP 800 22

The NIST FIPS SP 800 22 [20] lists 15 demanding tests, most of them inherited from DieHard, some requiring 1 Mbit of data, and others up to 1 Gbit.

D. NIST FIPS SP 800 90B

The standard NIST FIPS SP 800 90B [15] is methodological. In particular, it aims at understanding whether some implicit hypotheses assumed by other analyses are true in practice. For instance, the IID (Independent and Identically Distributed) test is prescribed. It helps clarify the reason for tests failures, if any.

E. BSI AIS 31

The BSI innovates in AIS 31 [12] mostly by requiring the tests to be performed on the randomness source in addition to the TRNG output. It also introduces the notion of *stochastic model* in section 2.4.1. Regarding the statistical tests, there are 9 of them. They require only $20,000$ bits (like NIST FIPS 140-2, recall Sec. II-B) to yield interesting results.

- Test T0 (disjointness test): birthday paradox for sub-strings.
- Test T1 (Monobit Tests): same as Frequency (Monobit) Test.
- Test T2 (same as Poker test)
- Test T3 (same as runs test)
- Test T4 (long run test)
- Test T5 (autocorrelation test)
- Test T6 (uniform distribution test)
- Test T7 (comparative test for multinomial distributions aka 'test for homogeneity')
- Test T8 (entropy estimation)

F. ISO/IEC 20543 draft

The abovementioned tests are either ad hoc (DieHard) or national standards (e.g., USA for NIST documents, Germany for AIS 31, etc.). Notice that other countries also emit recommandations, such as the French RGS (*Référentiel Général de Sécurité*).

For this reason, an international standardization project has been launched. The ISO/IEC 20543 is an ISO project [18] aiming (in particular) at unifying NIST FIPS SP 800 90C and BSI AIS 31. It insists on the *method*, but stresses that tests are required. In addition, it does require rationale evidence through the documentation of *stochastic models*.

III. Comparison and Extrapolation

A. Comparison of tests

A short comparison between AIS 31 and NIST SP 800-22 is given in [12, §2.4.5.1.].

A more complete comparison is provided in Tab. I. It can be seen that standards depend on the number of bits they require to compute a metric, and also about the number of tests. Still, we underline that each test can be adapted to apply to different sequence sizes, provided formulas are adapted. The more up-to-date tests are NIST FIPS SP 800 22 and BSI AIS 31.

B. Boolean functions

Tests described in Sec. II are based on some empirical properties on bit sequences. However, the evaluation of the strength of Boolean sequences has been extensively studied in another field, namely the Boolean functions analysis, which is typically studied in the reference document by Claude Carlet [5]. A Boolean function f is an application matching $\mathbb{F}_2^n$ to $\mathbb{F}_2$, where $\mathbb{F}_2 = \{0,1\}$ is the set of bits and n is a positive integer. The Cartesian product $\mathbb{F}_2^n$ refers to $\underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{n \text{ times}}$.

The evaluation criteria actually refer to some attacks, hence are specially suited for random bit sequences. They are listed below:

1) *non-linearity* ($nl(f)$): see [6, Definition 5, page 16],
$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbb{F}_2^n} \left| \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus u \cdot x} \right|$$
which shall be maximized in order to resist better attacks based on linear approximations;
2) *differential uniformity* ($\Delta(f)$): see [6, §3.1.3, page 28],
$$\Delta(f) = \max_{(u,v) \in (\mathbb{F}_2^n)^* \times \mathbb{F}_2} |\{x \in \mathbb{F}_2^n : f(x) \oplus f(x \oplus u) = v\}|$$
which shall be minimized in order to resist better attacks based on differential approximations;
3) *algebraic degree* ($d^\circ(f)$): see [5, §2.1, page 12]: let us denote $f(x) = \bigoplus_{u \in \mathbb{F}_2^n} a_u \ x^u$ the Algebraic Normal Form (ANF) of f [5, §2.1, page 9] (where $\forall u, a_u \in \mathbb{F}_2$); then
$$d^\circ(f) = \max \{|u| : a_u \neq 0\}$$

Table I
COMPARISON BETWEEN SOME DIFFERENT STANDARDS ABOUT TRNGS

Test	Ref.	Publication year	Number of tests	Sequence size (bits)
DieHard	[14]	1995	15	$\geq 80,000,000$
NIST FIPS 140-2	[21]	2001	4	$20,000$
NIST FIPS SP 800 22	[20]	2010	15	$\approx 1,000,000,000$
NIST FIPS SP 800 90B	[15]	2015	methodological	N/A
BSI AIS 31	[12]	2011	9	$20,000$
ISO/IEC 20543	[18]	2018	methodological	N/A

which shall be maximized for f to depart from linear functions;

4) *algebraic immunity*: see [5, §4.1.6]. Actually, the algebraic degree can be seen as a less mandatory criterion than the three others: the high-order differential attack is known to be efficient only for the second degree.

For more context on those metrics, we refer the reader to [5], [6], or [17, §3.1].

Notice that tests defined in current standards (recall Sec. II) are mostly invariant by block-wise permutations in the analyzed sequence of bits. Accordingly, this is reflected by Boolean function metrics, which do not change when the input bits are shuffled, hence an addition motivation to consider Boolean function metrics as valuable tests.

Notice that these metrics are difficult to compute on Boolean functions with many inputs n: one megabit is typically representing the truth table of a Boolean function with $n = 20$ variables.

IV. ATTACKS

As underlined in Fig. 1, a TRNG can also be operated in abnormal situations. Standardized test procedures usually avoid these cases, as they are hard to prescribe. Typically, quoting ISO/IEC 20543 ([18, §7.2.4]):

> ≪ *The test result shall be collected at representative environmental conditions inside the normal operating range (e.g. 25°C, 0°C, +100°C for temperature). To the extent that the device itself is not capable of detecting excursions from the normal operating range, it shall be ensured by operational guidance that the device is not subjected to conditions outside the regime so indicated.* ≫

However, attackers are inclined to push the product outside of its nominal operational environment. Therefore, an analysis of the TRNGs under stress is required in practice while evaluating attack paths.

There is a multiplicity of different TRNGs. In this section, we focus on one technology which is widely deployed: the ring-oscillator TRNG. The principle is depicted in Fig. 2, where the amplifier has 50 W power and the electromagnetic probe has length 30 mm, diameter 10 to 200 μm.

A. Example of harmonic injection attacks

Harmonic injection is a well known attack to force a ring-oscillator TRNG oscillate in an externally forced frequency

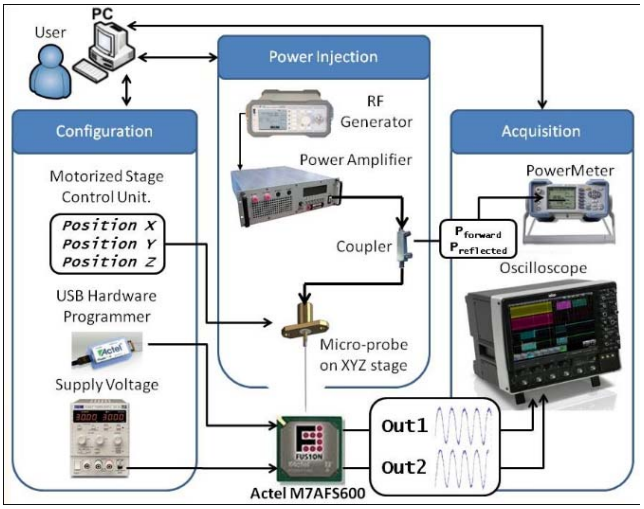

Figure 2. Principle of a ring-oscillator (RO) based TRNG forced by a strong EM field (courtesy of Pierre Bayon et al. [3])

operation. Actually, the oscillators making up the entropy source (recall Fig. 2) behave harmonically, i.e., have a periodic behavior. Ideally, that is without noise, the ring-oscillator has a periodic behavior. The intrinsic frequency of the ring-oscillator can be forced externally, by a strong and focused field, which will drive the oscillator in a mode which departs from its natural oscillation frequency. Amongst the various examples of external attacks with strong coupling, we can mention [13], [3], [2].

This principle has been used constructively to detect an approaching electromagnetic probe, interpreted as the preparation of an attack [9].

Besides, the same principle of influencing the frequency of a RO has been shown practical by injecting through the powerline and even through other connections (serial port, RS232, etc.) [22].

B. Hints about TRNGs failing

The attack described in Sec. IV implies an external adversary trying to bias the TRNG. In this section, we would like to address a more subtle situation whereby the coupling is actually carried out... by the device itself.

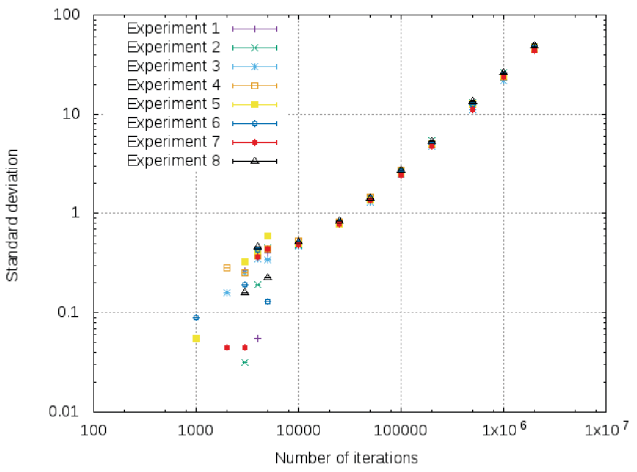

Figure 3. Standard deviation of the number of rotations in a loop-based TRNG versus the time to take the measurement

We carry out experiments on an ASIC designed in 40 nm CMOS technology. Those are reported in Fig. 3. We observe that for 8 independent experiments, the variance in the oscillating frequency of a ring-oscillator is heavily impacted by the noise when the oscillation time is short (between 1,000 and 10,000 clock cycles). Indeed, the system is starting fresh and is thus very sensitive to external noise.

However, after a certain number of iterations, the system has been coupled long enough to its environment (through cross-talk, substrate noise, etc.), hence operates in a mode which is consistent with the environment own resonating frequencies.

Indeed, let N_1 and N_2 be two random variables which obey the same normal law $\mathcal{N}(0, \sigma^2)$. If N_1 and N_2 are independent, then the variance of $N_1 + N_2$ is $2\sigma^2$. But if $N_1 = N_2$, then the variance of $N_1 + N_2$ is even larger, as it is equal to $4\sigma^2$. Actually, for an interaction time greater than $10,000$, all experiments (with results depicted in Fig. 3) yield a standard deviation growing linearly with the number of iterations. Hence the exploitable entropy vanishes, since the variance is not accountable on fresh entropy. Rather, the variations are residual uncertainties, and a better way to collect entropy is merely to restart from scratch the oscillations.

The same note has been observed in [8], where ring oscillators are characterized to be PUFs.

This situation is thus paradoxical in that the devil is actually already with the circuit: the coupling is favoring a degradation of the generated entropy.

Notice that a similar effect has been put forward regarding side-channel analysis: some protections involve the sharing of sensitive variables into independent random variables, which need to be recombined to reveal the information to protect. Some of the coupling effects (mentioned above) induce so-called *glitches*, which combine shares *constructively*, thereby ruining the intended effect of masking.

As a mitigation, it is therefore recommended to reseed the loop-based TRNG periodically, as done in TERO (Transient Effect Ring Oscillator) structures [23].

V. Conclusion

The quality of TRNGs can be assessed either by tests or by evaluation methodologies. Tests have the nice feature to be unambiguous (that is, they are enforceable against third parties). However, there is no consensual test, at least with deep roots in the scientific research field. In this paper, we list the most known tests and highlight the two more relevant ones (NIST FIPS SP 800-22 and BSI AIS 31). The BSI AIS 31 is more efficient because it is able to decide of the entropy quality based on only $20,000$ bits, that is several orders of magnitude less than the amount of random bits required to apply all the test from NIST FIPS SP 800-22. Besides, we advocate that traditional metrics aimed at characterizing Boolean functions can be applied to test random bit sequences too.

TRNGs are supposed to continue to deliver high quality random bitstrings even in adversarial conditions. However, it has been highlighted that a strong coupling forced from the outside of the TRNG could lock it, thereby having it fail most tests. We investigate in this article a phenomenon whereby self induction covers the noise, rendering the TRNG immune to external noise. Therefore, the TRNG becomes out of function as it is behaving deterministically. This means that the attacker happens to be the circuit itself! Hopefully, along the lines of defense in depth strategy, heterogeneous sensors (such as frequency, reset, power integrity verification) can complement the protection of TRNGs.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00399, Study on secure key hiding technology for IoT devices [KeyHAS Project]).

References

[1] Halak Basel. Physically Unclonable Functions — From Basic Design Principles to Advanced Hardware Security Applications. DOI: 10.1007/978-3-319-76804-5.

[2] P. Bayon, L. Bossuet, A. Aubert, and V. Fischer. Electromagnetic analysis on ring oscillator-based true random number generators. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 1954–1957, 2013.

[3] Pierre Bayon, Lilian Bossuet, Alain Aubert, Viktor Fischer, François Poucheret, Bruno Robisson, and Philippe Maurine. Contactless electromagnetic active attack on ring oscillator based true random number generator. In *Proceedings of the Third international conference on Constructive Side-Channel Analysis and Secure Design*, COSADE'12, pages 151–166, Berlin, Heidelberg, 2012. Springer-Verlag.

[4] Christoph Böhm and Maximilian Hofer. Physical Unclonable Functions in Theory and Practice, 2013. DOI: 10.1007/978-1-4614-5040-5.

[5] Claude Carlet. Boolean Functions for Cryptography and Error Correcting Codes: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering. pages 257–397. Cambridge University Press, Y. Crama and P. Hammer eds, 2010. Preliminary version available at http://www.math.univ-paris13.fr/~carlet/chap-fcts-Bool-corr.pdf.

[6] Claude Carlet. Vectorial Boolean Functions for Cryptography: Chapter of the monography Boolean Models and Methods in Mathematics, Computer Science, and Engineering. pages 398–469. Cambridge University Press, Y. Crama and P. Hammer eds, 2010. Preliminary version available at //www.math.univ-paris13.fr/~carlet/chap-vectorial-fcts-corr.pdf.

[7] Heebong Choi. ISO/IEC 20540. Information technology – Security techniques – Guidelines for testing cryptographic modules in their operational environment. ISO/IEC JTC1/SC 27/WG3. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=68294.

[8] Robert Hesselbarth, Florian Wilde, Chongyan Gu, and Neil Hanley. Large scale RO PUF analysis over slice type, evaluation time and temperature on 28nm Xilinx FPGAs. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2018, Washington, DC, USA, April 30 - May 4, 2018*, pages 126–133. IEEE Computer Society, 2018.

[9] Naofumi Homma, Yu-ichi Hayashi, Noriyuki Miura, Daisuke Fujimoto, Daichi Tanaka, Makoto Nagata, and Takafumi Aoki. EM Attack Is Non-invasive? - Design Methodology and Validity Verification of EM Attack Sensor. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems – CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg, 2014.

[10] Yohei Hori, Takahiro Yoshida, Toshihiro Katashita, and Akashi Satoh. Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs. *Reconfigurable Computing and FPGAs, International Conference on*, 0:298–303, 2010.

[11] ISO/IEC JTC1/SC27/WG3. ISO/IEC 19790. Information technology – Security techniques – Security requirements for cryptographic modules, 2012. https://www.iso.org/standard/52906.html.

[12] Wolfgang Killmann and Werner Schindler. A proposal for: Functionality classes for random number generators, September 2011. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile.

[13] A. Theodore Markettos and Simon W. Moore. The Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators. In Christophe Clavier and Kris Gaj, editors, *CHES*, volume 5747 of *Lecture Notes in Computer Science*, pages 317–331. Springer, 2009.

[14] George Marsaglia. The Marsaglia random number CDROM including the Diehard Battery of Tests of randomness. Web site at the Department of Statistics, Florida State University, Tallahassee, FL, USA., 1995.

[15] NIST. Recommendation for the entropy sources used for random bit generation, 2012. http://csrc.nist.gov/publications/drafts/800-90/draft-sp800-90b.pdf.

[16] NIST. Proposed Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process, August 1st 2016. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-draft-aug-2016.pdf.

[17] Gilles Piret, Thomas Roche, and Claude Carlet. PICARO – A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS*, volume 7341 of *Lecture Notes in Computer Science*, pages 311–328. Springer, 2012.

[18] Jean-Pierre Quémard, Hans von Sommerfeld, and Randall Easter. ISO/IEC 20543. Information Security – Security Techniques – Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408. 1st draft of IS. ISO/IEC JTC1/SC 27/WG3. http://www.iso.org/iso/fr/home/store/catalogue_tc/catalogue_detail.htm?csnumber=68296.

[19] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 237–249. ACM, 2010.

[20] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications,, april 2010. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf.

[21] NIST FIPS (Federal Information Processing Standards). Security Requirements for Cryptographic Modules publication 140-2, May 25 2001. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf.

[22] Takeshi Sugawara, Yu ichi Hayashi, Naofumi Homma, Takaaki Mizuki, Takafumi Aoki, Hideaki Sone, and Akashi Satoh. Mechanism behind Information Leakage in Electromagnetic Analysis of Cryptographic Modules. In *WISA*, volume 5932 of *Lecture Notes in Computer Science*, pages 66–78. Springer, August 25-27 2009. Busan, Korea. DOI: 10.1007/978-3-642-10838-9_6.

[23] Michal Varchola and Miloš Drutarovský. New high entropy element for fpga based true random number generators. In *Proceedings of the 12th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'10, pages 351–365, Berlin, Heidelberg, 2010. Springer-Verlag.

APPENDIX A
COMPARISON BETWEEN TRNGS AND PUFS

Physically Unclonable Functions (PUFs, see e.g. [1], [4]) are being standardized, in international projet ISO/IEC 20897. The notion of entropy and the means to test it are central to the procedure of PUFs security requirements test and evaluation.

A. Introduction on PUFs

The responses from multiple PUFs are arranged into a cube as Fig. 4 shows. The repetitive calls to a single PUF are illustrated in Fig. 5. The single small cube describes a 1-bit response from a PUF. The three axes of the cube and the time are described hereafter, as directions:

- **direction B**: "#bit" shows the bit length of the response obtained from a single challenge. In a 1-bit response PUF, e.g., arbiter PUF, the dimension B collapses.
- **direction C**: "#challenge" shows the number of different challenges given to a PUF. In a no-challenge PUF (or, more rigorously, a one-challenge PUF), e.g., SRAM PUF, the dimension C collapses.
- **direction D**: "#PUF" shows the number of different PUF devices under test.
- **direction T**: "#query" shows the number of query iterations under the fixed PUF device and challenge.

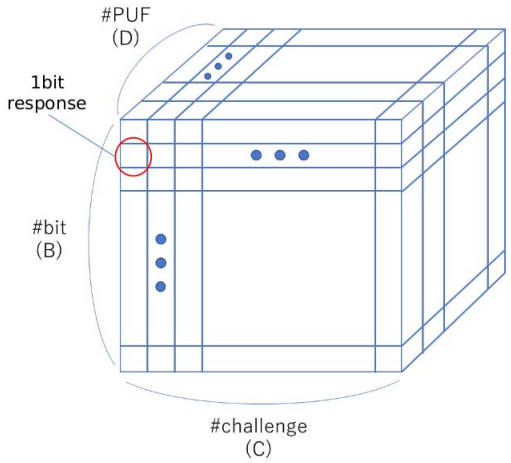

Figure 4. The three dimensions involved in the PUFs entropy metrics (courtesy of Hori [10])

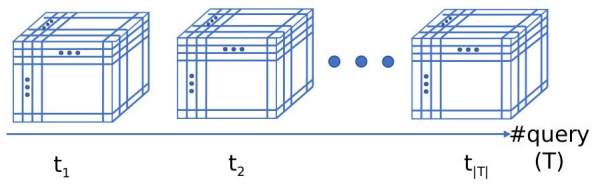

Figure 5. Illustration of the PUF repetitive usage, involved in the PUF steadiness metric (courtesy of Hori [10])

B. List of metrics (security requirements)

The definition and the explanation of security requirements for PUFs are given here-after:

- **Steadiness**[1]: it is a measurement of stability of PUF responses in time. This metric can be seen as a *safety* requirement. However, PUFs with unsteady responses could be prone to prediction attacks (if the response is very biased) or to related keys attacks.
- **Randomness**: it assesses how unpredictable are PUF responses when considering a collection of response bits under all the possible challenges. The obtained *intra-PUF* bitstring shall be, ideally, unpredictable. Such security requirement attests of the PUF unclonability.
- **Uniqueness**: it estimates how different are any two pairs of different PUFs. This *inter-PUF* metric is required to quantify in which respect the fab is unable to generate clones of PUFs.
- **Unpredictability**: it estimates how hard it is to predict the responses of an $(n+1)^{\text{th}}$ PUF knowing all previous n instances. This metric relates to randomness, but is more pragmatic as it involves machine learning or *ad hoc* tests.
- **Unclonability**: this metric makes sure no easy exploitable bias exist in the PUF architecture, by design. The goal of this security requirement is to valide for the absence of trap or backdoor in the PUF rationale.

C. Tests and evaluations

The security requirements defined in Sec. A-B have some value only provided one has a means to attest that a device indeed implements them. There are two methods to do so: *test* and *evaluation*.

In testing philosophy, an automatic procedure is launched to check each and any security requirement. This allows for fast and reproducing checking. However, subtle issues (e.g., corner cases, weak vulnerabilities, problems not covered by the test suite, etc.) could inadvertently pass successfully through the test.

This explains why testing is complemented by the evaluation philosophy. Evaluation is conducted by an expert, who attempts to think out-of-the-box in a view to derive attacks. This expert defines a couple of attack paths, performs a quotation (i.e., scores which reflect the cost of the attacks) for them, selects and realizes the attack of lowest quotation.

[1]Notice that *steadiness* is a word reserved for stability of a given PUF response corresponding to a fixed challenge. The synonymous terms *reliability*, *reproducibility* and *stability* are not preferred. In particular, "reliability" is discarded as it would make some confusion regarding the metric related to the yield in the CMOS manufacturing processes.

D. Well established tests

The tests base themselves on a *metric*. For the tests to be consistent even in heterogeneous conditions, the metric must be generic. The idea is that the metric must suit to a variety of PUFs.

Entropy is a metric which can compare data of various nature. The output of discussion at standardization committee meetings is to use this very same metric for different security requirements. The method is termed "**multiple data, one same metric**". This enables consistency within metrics, and simplifies the test of security requirements. Notice that the entropy for the steadiness is simply $H_2(\text{BER}) \triangleq -\text{BER}\log_2\text{BER} - (1-\text{BER})\log_2(1-\text{BER})$, where BER is the Bit Error Rate.

E. Well established evaluations

Evaluation is required for those security requirements which cannot be tested, because they cannot be decided based on measured data. This happens for requirements which are "*negative*": this means that the security requires *not to* verify a property.

This holds for instance for "unclonability". One aspect of unclonability could be termed "supervised" unclonability: the attacker manages to predict responses from unseen challenges, after having seen enough responses from known challenges. This metric, termed *mathematical unclonability*, can be estimated as the entropy in the C-B space, hence can be turned into a test.

However, *physical unclonability*, consists in evaluating the difficulty of fabricating a PUF that has the same CRPs as a specific PUF. This task can only be achieved by a thorough qualitative analysis of the PUF design.

1) Further tests and evaluations: The tests described in Sec. A-D and the evaluations described in Sec. A-E are natural. Still, some more "specialized" (if not bespoke) methods can be leveraged for challenging more drastically the security requirements.

Regarding tests, entropy can also be *characterized* (instead of *estimated*) by some testsuites. Still, it is important those tests adhere to the principle: **multiple data, one same testsuite** (e.g., FIPS SP 800-22, FIPS 800-90B, AIS31, etc., see section II)

Still regarding tests, machine learning (ML) of challenge and response pairs has been shown to be able to predict with good accuracy responses from unseen challenges [19]. Thus, the general-purpose entropy metric can be traded for crafted tools, e.g., using ML, or any taylored distinguisher. Still, for consistency reasons, this analysis shall adhere to the principle: **multiple data, one same distinguisher**. For the sake of clarification, the equivalent of *randomness* when trading entropy for ML tools is referred to as *unpredictability* (or alternatively: *mathematical unclonability*).