# Characterization of statistical anomalies at the output of a random number generator

Antoine Levotre

THÈSE

Pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES**

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques
Unité de recherche : Institut Fourier

# Caractérisation d'anomalies statistiques en sortie d'un générateur de nombres aléatoires

# Characterization of statistical anomalies at the output of a random number generator


Présentée par :

**Antoine LEVOTRE**

Direction de thèse :

| | |
|---|---|
| **Philippe ELBAZ-VINCENT**<br>PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES | Directeur de thèse |
| **Cécile DUMAS** | Co-encadrante de thèse |

Rapporteurs :

**DAVID LUBICZ**
SENIOR SCIENTIST, DGA MAITRISE DE L'INFORMATION
**DAMIEN VERGNAUD**
PROFESSEUR DES UNIVERSITES, SORBONNE UNIVERSITE


Thèse soutenue publiquement le **6 mai 2024**, devant le jury composé de :

| | |
|---|---|
| **DAVID HELY,**<br>PROFESSEUR DES UNIVERSITES, GRENOBLE INP | Président |
| **PHILIPPE ELBAZ VINCENT,**<br>PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES | Directeur de thèse |
| **DAVID LUBICZ,**<br>SENIOR SCIENTIST, DGA MAITRISE DE L'INFORMATION | Rapporteur |
| **DAMIEN VERGNAUD,**<br>PROFESSEUR DES UNIVERSITES, SORBONNE UNIVERSITE | Rapporteur |
| **VIKTOR FISCHER,**<br>PROFESSEUR DES UNIVERSITES EMERITE, UNIVERSITE DE SAINT-ETIENNE - JEAN MONNET | Examinateur |
| **JEAN-CLAUDE BAJARD,**<br>PROFESSEUR DES UNIVERSITES EMERITE, SORBONNE UNIVERSITE | Examinateur |
| **LAURENT-STEPHANE DIDIER,**<br>PROFESSEUR DES UNIVERSITES, UNIVERSITE DE TOULON | Examinateur |

Invités :

**CECILE DUMAS**
DOCTEUR EN SCIENCES, CEA CENTRE DE GRENOBLE

# Remerciements

Tout d'abord j'aimerais remercier l'ensemble des membres du jury pour avoir accepté d'examiner cette thèse, et plus particulièrement David Lubicz et Damien Vergnaud pour le temps qu'ils auront dédié à rapporter le manuscrit, et pour leurs précieux commentaires.

Je remercie également l'Institut Fourier ainsi que le CEA Grenoble pour le financement de ma thèse et les moyens conséquents mis à ma disposition tout au long de ces trois ans, qui m'auront permis d'explorer avec passion ce sujet (en témoignent les nombreux graffitis sur le tableau du bureau thésards).

Et puisque l'on mentionne le bureau des thésards, j'aimerais remercier tous mes collègues, et néanmoins amis du CESTI, à commencer par les nombreux doctorants et alternants qui auront partagé avec moi la 338, Gabriel, Vincent, Damien, Dorian et Valentin, mais également tous les autres membres du laboratoire avec qui j'ai partagé ces trois années. Un grand merci tout particulier à Anne pour son accueil et sa bienveillance, malgré des heures passées à m'entendre chantonner le générique des Cités d'or.

Enfin, impossible de ne pas remercier Cécile et Philippe pour leur suivi quotidien tout au long de ces trois ans. Ce serait un euphémisme de dire que cette thèse n'aurait pas porté les mêmes fruits sans leur confiance pour me laisser pousser jusqu'au bout mes idées.

## Résumé en français

La génération de nombres aléatoires est l'une des pierres angulaires de la cryptographie. Celle-ci est utilisée dans la génération de clés secrètes, de nonces, ou dans l'implémentation de contre-mesures. Pour assurer la sécurité des mécanismes cryptographiques, il est nécessaire de disposer d'un aléa de « bonne qualité ». En particulier, il ne doit pas être possible d'obtenir des informations sur les nombres futurs ou précédemment produits par un générateur de nombres aléatoires (abrégé RNG).

Pour évaluer la qualité d'un RNG, plusieurs méthodologies sont envisageables selon le type de générateur, comme par exemple l'utilisation de mesures statistiques sur les données produites par celui-ci. Ces mesures sont ensuite utilisées dans des tests d'hypothèses, aussi appelés « tests statistiques », qui ont pour vocation de rendre une décision binaire sur l'acceptation ou non de l'hypothèse de test. Dans le cas de la génération d'aléa, cette hypothèse, est le plus souvent que les données étudiées peuvent avoir été générées par un générateur idéalement aléatoire. Bien que les tests statistiques soient couramment utilisés pour évaluer la qualité d'un RNG cette méthodologie présente plusieurs inconvénients. Tout d'abord, l'aspect binaire de la décision du test ne permet pas de rendre compte de l'amplitude de l'échec ou de la réussite de celui-ci. Ensuite, selon la statistique utilisée, l'échec d'un test ne permet pas nécessairement de remonter à la caractéristique statistique de la séquence faisant défaut. Ceci pourrait pourtant intéresser le concepteur d'un RNG qui souhaite utiliser les tests statistiques comme moyen d'affiner l'architecture ou le paramétrage de son générateur.

Dans cette thèse, nous allons donc nous pencher sur le développement de modèles statistiques liés à deux caractéristiques de grande importance dans la génération d'aléa binaire : la fréquence d'apparition des bits 0 et 1, et la corrélation entre les bits d'une séquence. A partir de ces modèles, nous proposons ensuite deux statistiques de tests comme outils d'évaluation plus fins de ces deux caractéristiques précises. Dans le cas de l'étude des corrélations, le test que nous proposons s'inscrit en remplacement de l'un des tests les plus utilisés aujourd'hui, appelé test d'autocorrélation, et présent notamment dans la batterie de tests du standard allemand AIS 20/31. En effet ce dernier a pour défaut majeur de pouvoir échouer même lorsque les bits de la séquence étudiée ne présentent pas de corrélation, défaut qui est corrigé par notre test.

Dans la suite de la thèse, nous proposons d'appliquer notre statistique d'étude des corrélations à un nouveau cas d'usage : l'attaque d'un RNG. L'attaque que nous présentons doit, si

elle est réussie, induire de la corrélation dans les données, mais celle-ci requiert un paramétrage relativement précis du banc d'attaque pour fonctionner. Nous montrons donc comment l'application de notre statistique peut permettre d'automatiser une partie du processus d'attaque et faciliter en ce sens le travail d'un attaquant.

Enfin, dans la dernière partie de la thèse, nous affinons davantage notre statistique d'étude des corrélations dans le but de rendre plus précise l'étude des séquences par le biais de cette statistique. Pour cela, nous proposons deux extensions de notre premier modèle de corrélation : un modèle linéaire déjà existant, nommé « autocorrélation partielle », et un modèle multiplicatif que nous avons développé. La statistique issue du modèle multiplicatif est plus simple d'écriture mais bien plus complexe en temps de calcul et en mémoire que la fonction d'autocorrélation partielle. En définitive, nous préconisons donc l'usage de la fonction d'autocorrélation partielle (qui se repose sur un usage préalable de notre première statistique d'autocorrélation) pour évaluer la présence de corrélations au sein de données binaires.

# Contents

CHAPTER 1

# Introduction

Random number generation is the cornerstone of modern cryptographic mechanisms, playing a major role in the generation of secret keys, ephemeral data such as nonces, or in the implementation of countermeasures such as the masking of sensitive data. A weakness in the quality of the randomness produced by a generator can compromise the security of a theoretically reliable cryptographic scheme, and therefore of the system on which it is based. In [**60**] and [**61**], Nguyen and Shparlinski explain, for example, that knowledge of a few bits of successive nonces of DSA and ECDSA enables an attacker to trace back to the private signature key, thus proving the need for nonces to be unpredictable. More recently, Ebalard and Benadjila also demonstrated at the SSTIC conference in June 2023 [**25**] that some Cisco ASA equipment using an ECDSA signature mechanism suffered from a nonce duplication, due to a weakness in the random number generator. This duplication led to the recovery of the private keys (CVE-2023-20107, CSCvm90511), and thus to a total compromise of the equipment's security. A number of similar attacks have occurred on various systems, such as the breaking of RSA keys for SSH/SSL on OpenSSL in 2008, or the breaking of the Playstation 3 signature key in 2010 [**27**] (slides 122-130), as presented in the introduction to the presentation [**25**]. Similarly, Elbaz-Vincent and Traoré [**26, 77**] analyzed hundreds of thousands of RSA X.509 certificates [**19, 81**], and found several certificates which shared the same RSA moduli, potentially due to the use of a defective random prime generator. From all those examples, it appears critical to have methods at hand to ensure the quality of the randomness produced by a random number generator.

To assess the quality of the random numbers produced by a generator, several approaches are possible. For a generator based on an intrinsically random *physical phenomenon*, the most robust and currently preferred method ([**6**] §3.2.2 (3), [**43**] §4.5.3 (319), [**67**] §4.5 (613)) is to find the stochastic model of the generator, i.e. a probabilistic model describing the consecutively generated numbers, based on the physical model of the phenomenon used. From the stochastic model we can then derive an expression for the entropy [**73**] of the generator, an image of the quality of the randomness. In the rest of the manuscript, we will denote this type of generator by the abbreviation **TRNG**, or **PTRNG** for *True Random Number Generator* or *Physical True Random Number Generator* respectively.

In the case of a *Deterministic Random Number Generator*, abbreviated in this manuscript as **DRNG**, which relies on a *deterministic algorithm* to produce sequences of numbers, it is possible to obtain an assurance on the quality of the randomness produced by formally proving

that the algorithm used produces uniformly random data. In most case however, this assurance on the quality rely on the hypothesis that the internal state (internal variables, such as the keys) of the algorithm is kept secret.

A third class of random number generators, called *Hybrid Random Number Generators*, combines a TRNG and a DRNG. More specifically, a TRNG produces random inputs for a DRNG, the latter's role being to eliminate the potential imperfections in the randomness produced by the TRNG.

When the stochastic model for a TRNG or the formal proof for a DRNG are not available, or in the case of a generator that is non-deterministic but not based on a physical phenomenon, the evaluation of the quality of the randomness must involve the use of "black-box" statistical tests on the generated sequences. The goal of a statistical black-box test is to study the statistical properties of sequences, without any prior knowledge on the generator (hence the term *black-box*).

Among the batteries of statistical tests considered are those of the **NIST**[1] in the United States (SP 800-90 B [**6**]), the **BSI**[2] in Germany (procedures A and B of AIS 20/31 [**43**]), which are designed to compare the characteristics of sequences to those of a theoretical ideally uniform sequence. In particular, we study the characteristics of uniformity (proportion of 0s and 1s, or distribution of multi-bit patterns) and the independence of successive bits or patterns.

More specifically, the statistical tests proposed by AIS 20/31 and SP 800-90 B are hypothesis tests with the following null hypothesis: "The tested sequence is likely to have been produced by an ideal generator". Each statistical test returns a numerical value which, depending on the deviation from the expected value for an ideally random sequence, allows for a decision to be made about whether to accept or reject the null hypothesis.

One of the drawbacks of this test methodology is that, beyond the potential errors of type I (erroneous rejection of the null hypothesis) or II (erroneous acceptance of the null hypothesis), the simple rejection of the null hypothesis does not make it possible to distinguish the characteristics of the faulty tested sequence. Sometimes, as we will show later, a statistical test can fail even though the tested sequence is not impacted by the anomaly the test aims at covering.

In addition to the evaluation of the quality of the randomness produced by a generator when it functions properly, statistical tests can also be used to detect a failure of the generator due to an attack, or due to aging (see [**43**] §4.2.2 (289)) for example. These tests, called *total failure* and *on-line* tests, aim at detecting at total loss of entropy (no randomness in the data), or non-tolerable statistical defects which may suddenly occur during the generation process

[1] *National Institute of Standards and Technologies*
[2] *Bundesamt für Sicherheit in der Informationstechnik*

respectively.

In France, the ANSSI[3] [**1**] proposes a number of rules and recommendations on the evaluation of a physical random number generator, but these are far less restrictive than their NIST or BSI counterparts concerning the "internal" part of the generator (which produces the numbers that are meant to be post-processed). Indeed, the French agency ([**1**] §2.4.2) simply imposes the following rules, which we propose to translate as such: a "*functional description*"[4] of the generator must be provided and "*statistical tests on the output of the physical generator must not show any significant statistical defect in the produced randomness*",[5] without any restriction on the tests that must be applied, or on what a "*significant defect*" must be defined as. However, the ANSSI puts the stress on the necessity of having a good post-processing algorithm, based on the assumption that rigorously proving the quality of a post-processing is much easier than proving the quality of a physical source of entropy (see [**1**] *Note* in p.31 and RègleArchGDA in §2.4.1).

In this manuscript, we will then start by taking a closer look at both deterministic and non-deterministic random number generation methods, with a particular focus on current standards on this matter. This will be the subject of Chapter 2.

In Chapter 3, we will review the various methods to evaluate the quality of the randomness produced by these generators. We will again discuss current standards, particularly in terms of black-box statistical testing, and we will discuss the limits of this evaluation methodology. We will also provide an example of stochastic modeling for a given TRNG to illustrate how the model can be used to provide a measure of the quality of the randomness supplied by the generator.

Chapters 4 and 5 will be devoted to the development of probabilistic models to characterize two given statistical anomalies: the global disproportion of bits 0 and 1 in a sequence, and the correlation between successive bits. We will thus provide a precise definition of a "statistical defect" in a random sequence. In Chapter 6, we will observe how the two models interact with each other. A new statistical test, introduced in Chapter 7, will then derived from the correlation model, and addresses one of the problems of a standard test currently used.

In Chapter 8, we will use our newly-developed test in a different context, namely the detection of the success of an attack on a TRNG. We will see how the test can be used to observe the impact of an attack on the statistical properties of generated sequences.

Finally, in Chapter 9, we will explore the possibility of extending our model developed in Chapter 5 to characterize more precisely and directly the correlation phenomena between

[3] *Agence Nationale de la Sécurité des Systèmes d'Information*

[4] "Le générateur physique d'aléa doit disposer d'une *description fonctionnelle*." [**1**] §2.4.2, RègleArchiGVA.1

[5] "Des tests statistiques en sortie du générateur physique ne doivent pas faire apparaître de défauts significatifs dans l'aléa généré.", [**1**] §2.4.2, RègleArchiGVA.2

bits of a given sequence.

And chapter 10 will conclude this manuscript, summarizing our key contributions, and offering some leads on how to further formalize the analysis of statistical anomalies in sequences produced by random number generators.

CHAPTER 2

# Random number generation

In this chapter, we will set the mathematical ground that we will rely on through this whole manuscript to properly define random number generation from a theoretical standpoint.

We will then present practical implementations of random number generators, looking first at deterministic generators, and more specifically at the current standards on generation functions and the standard safety levels to which they are subject.

Next, we will look at random number generation based on physical phenomena, providing several examples of TRNGs. These will later be used to illustrate our points, but the list we provide is by no means exhaustive.

## 2.1. Probability and random variables

The following section will aim at summarizing different definitions proposed in [**47**]. In their book (§2.1), the authors introduce the probability theory as being "concerned with situations which may result in different outcomes". Mathematically, these different outcomes are points in a space $\mathcal{Z}$.

DEFINITION 2.1 ($\sigma$-field on $\mathcal{Z}$). *For a given set $\mathcal{Z}$, a $\sigma$-field $\mathscr{C}$, on $\mathcal{Z}$ is a non-empty collection of subsets of $\mathcal{Z}$ which is closed under complementation, and countable union.*

The *events* we are interested in are elements of a $\sigma$-field $\mathscr{C}$. This means in particular that any countable union of events is an event, and that the complementary of an event is also an event, which translates mathematically into:

$$\bigcup_{i\in I} C_i \in \mathscr{C}, \text{ with } C_i \in \mathscr{C} \text{ for all } i \in I, I \subset \mathbb{N},$$

where $\cup$ is the set union symbol, and

$$C^c = \mathcal{Z} - C \in \mathscr{C}, \text{ with } C \in \mathscr{C}.$$

DEFINITION 2.2 (Measurable space). *The couple $(\mathcal{Z},\mathscr{C})$, where $\mathscr{C}$ is a $\sigma$-field on $\mathcal{Z}$ is called a measurable space.*

DEFINITION 2.3 (Probability measure). *A probability measure on $\mathscr{C}$ is a function $\mu$ such that $\mu(\emptyset) = 0$, $\mu(\mathcal{Z}) = 1$ and such that, for $I \subset \mathbb{N}$ and $C_i$, $i \in I$ elements of $\mathscr{C}$:*

$$\mu\left(\bigcup_{i\in I} C_i\right) = \sum_{i\in I} \mu(C_i),\ \textit{if } C_i \cap C_j = \emptyset \textit{ for all } i \neq j.$$

*where* $\cap$ *is the set intersection symbol.*

In practice, when observing the results of random experiments, a quantifiable measurement of the events is desirable. Such a measurement is then the result of a function $T$ with values in a given space $\mathscr{T}$, which generates another $\sigma$-field $\mathscr{B}$ of sets $B$ such that:

$$C = T^{-1}(B) = \{z \mid z \in \mathscr{Z}, T(z) \in B\},$$

where $C \in \mathscr{C}$.

This space $\mathscr{T}$ is often equal to $\mathbb{R}$, and the function established in this case is called a *random variable*. More precisely, a random variable is defined as follows:

DEFINITION 2.4 (Random variable). *A random variable* $X : \mathscr{Z} \to \mathbb{R}$ *is a function which transforms outcomes in* $\mathscr{Z}$ *into observable values* $x \in \mathbb{R}$.

For $\mathscr{A}$ the $\sigma$-field of left-open intervals on $\mathbb{R}$ (intervals of the form $\{x \mid a < x \leqslant b, (a,b) \in (\mathbb{R} \cup \{-\infty, +\infty\})^2\}$), a random variable $X$ generates a probability measure $\Pr^X$ over $(\mathbb{R}, \mathscr{A})$ such that, for an event $A \in \mathscr{A}$:

$$\Pr^X(A) = \Pr(\{z \mid z \in \mathscr{Z}, X(z) \in A\})$$

where Pr is the probability measure over the outcome measurable space $(\mathscr{Z}, \mathscr{C})$ as previously defined.

DEFINITION 2.5 (Probability distribution of a random variable X). *For a random variable* $X$*, the probability measure* $Pr^X$ *over* $(\mathbb{R}, \mathscr{A})$ *described above is called the probability distribution of* $X$.

This formal definition of a random variable and of the underlying probability theory then enables us to properly define random number generation from a theoretical standpoint.

## 2.2. Theoretical concept of random number generation

In [**43**], the authors present random numbers as "realizations of random variables".[1]

A **random number generator** can thus be presented as a **system conducting successive random experiments**, with the outcome of each experiment (the drawing of a random number) being the **realization of a random variable**. In this particular context, the space of observable

[1] [**43**] §2.3.1 (124).

values is the set of possible outputs of the generator, usually equal to $\{0,...,2^n-1\}$ (the set $n$-bit integers) for generators used for cryptographic applications. In the rest of this section, we will simply denote this set by $\mathcal{V}$.

The random variables underlying to each successive generation of numbers can also be collectively seen as a single random process $\{X_t\}$ which is observed at different points in time $t \in \mathcal{T}$. The object $\{X_t\}$ defined as such is called a *stochastic process*.

More precisely, in the context of random number generation, the time space is countable, each new number being a distinct realization of the process. We then talk of $\{X_t\}$ as being a *discrete* stochastic process, and we denote it by $\{X_i\}_{i\in\mathbb{N}}$.

For a cryptographic system relying on random numbers, the random experiments conducted by an ideal random number generator must respect a certain number of rules, as stated in [**43**]. The experiments must be **unpredictable**, **independent**, and **unbiased**.

**Unpredictable** means that "*the observable outcome of the experiment is (to a certain extent) unknown before it is conducted*". In other word, an observer must not be able to anticipate the outputs of the generator with any significant advantage.

**Independent** means that the successive numbers must not have any influence on each other, and in particular that no advantage can be obtained on the knowledge of the future numbers based on the already generated ones.

And **unbiased** means that every number has the same probability of being generated at any given time.

This set of rules means that the underlying stochastic process $\{X_i\}$ of an ideal random number generator must also have some specific statistical properties. More precisely, the unpredictability is tied to the notion of entropy[2] [**73**]. The higher the entropy, the better the unpredictability of the generated numbers. We will further discuss how the notion of entropy can be used as a measure of the quality of the random numbers in section 3.1.

The need for independence of the random experiments means that, ideally, for any $(i_1,...,i_n) \in \mathbb{N}^n$, the random variables $X_{i_1},...,X_{i_n}$ must be independent. In other words, for any $(v_1,...,v_n) \in \mathcal{V}^n$, we must have:

$$\Pr\left(X_{i_1} = v_1,...,X_{i_n} = v_n\right) = \Pr\left(X_{i_1} = v_1\right) \times ... \times \Pr\left(X_{i_n} = v_n\right).$$

Finally, using the same notations as above, the "unbiased" criteria means that, for any $i \in \mathbb{N}$ and $v \in \mathcal{V}$:

[2] "*Entropy quantifies the amount of unpredictability relative to the observer.*" [**43**], §2.1.1 (72).

$$\Pr(X_i = v) = \frac{1}{|V|},$$

where $|\mathcal{V}|$ is the cardinal of the set of observable values $\mathcal{V}$. In this case, the random variable $X_i$ is said to be uniformly distributed.

In the following subsections, we will detail how random number generators are constructed in practice, in the case of both DRNGs (based on a deterministic algorithm) and PTRNGs (based on a physical phenomenon).

## 2.3. Random number generators

In his review of the history of random number generation, L'Ecuyer [**45**] explains that the use of random numbers dates back to at least 5000 years ago, in the form of dices.

Much more recently, in 1947, the RAND corporation [**13**] built the first fully automated random number generator, based on a random frequency pulse source, which was gated by a constant frequency pulse source, and served to increment 5-bit counter to provide integers between 0 and 31 at its output about once a second.

In practice, no matter the phenomenon or methodology used to produce random numbers, random number generation is divided in an **intrinsically random part**, used to produce random events and which can sometimes be external to the architecture of the generator itself, and of a **deterministic part** which produces a number based on the outcomes of each successive random event.

In the case of a dice roll for example, the random event is the roll itself, and the deterministic part is the transformation of the physical state of the dice into a number between 1 and 6 by reading the number of dots on the upper face once the dice stopped rolling. For the generator of the RAND corporation, the random frequency pulse which increments the 5-bit counter is the random part, and the deterministic part is the translation of the state of the counter into a number between 0 and 31 around once per second.

From these two examples we can already have a glimpse of the diversity of mechanisms that can be used to produce random numbers. In the following two sections, we will describe in more detail two classes of random number generators: deterministic random number generators (DRNGs), and non-deterministic random number generators, particularly generators whose random part is built around an intrinsically random physical phenomenon, i.e. the PTRNGs.

## 2.4. Deterministic random number generators

The aim of deterministic random number generators (or pseudorandom number generators) is always the same: to generate data sequentially using an deterministic algorithm, so that it

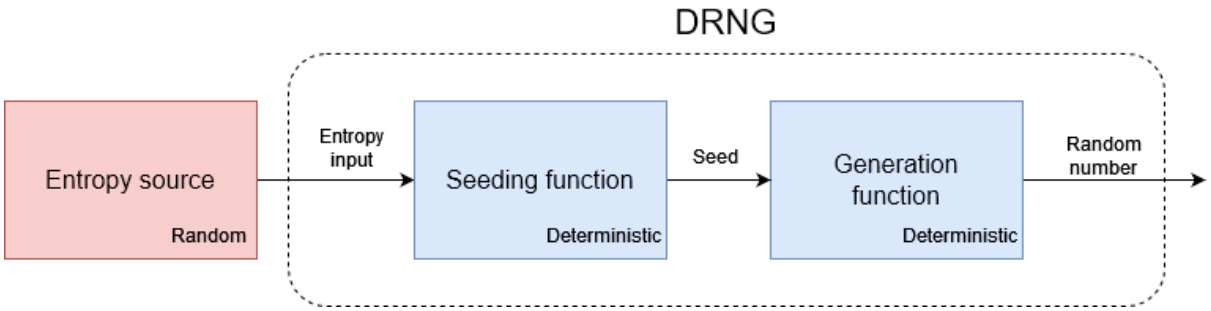



FIGURE 1. Schematic view of a generic DRNG.

is indistinguishable from ideally random data. To do this, an initial data, often called a *seed*, is instantiated using external random data called the *entropy input* in the SP 800-90A ([**4**] §7). The seed is then introduced as input to a function whose output is uniform over all the words in the output alphabet, in the sense that each word in the alphabet has the same probability to be the image of a random input. Optionally, a new seed can be reinstantiated during the course of the generation to "refresh" the randomness.

In this sense, the random part is the entropy source, and is external to the DRNG itself, and the deterministic part is the algorithm which produces the seed and random numbers from this seed (and from other parameters, as described for example in subsection 2.4.2). See Fig. 1 for the schematic representation of a generic DRNG.

To not compromise the unpredictability of the generated data, the seeds must ideally be impossible to influence, and must of course stay secret. In addition to this restriction on initial data, additional guaranties on the security of the generators must be established. The AIS 20/31 standard [**43**] lists four security properties, leading to four classes of generators, achieving increasing levels of security.

**2.4.1. Security properties of the AIS 20/31 standard.** The first listed property is called the *forward secrecy*, and states that a person with access to the set of generated numbers up to an instant $t$ must not be able to anticipate the data which will be generated after that instant $t$. The set of deterministic generators verifying the *forward secrecy* property corresponds to the first class of deterministic generators defined by the AIS 20/31 standard, named `DRG.1` (for *Deterministic Random Generator of class 1*).

Another security property is the *backward secrecy*. For this property to be verified, a person who has access to all the data generated after an instant $t$ must not be able to trace back (fully or partially) to the data generated before that instant. The generators verifying the properties of forward and backward secrecy correspond to the `DRG.2` class defined by AIS 20/31.

These two security properties can also be enhanced by assuming that, even with an access to the internal state of the generator at an instant $t$ in addition to the knowledge of the generated

data, one must not be able to compute the next values (for the *enhanced forward secrecy*) or the preceding values (for the *enhanced backward secrecy*). A DRG.2 generator which verifies the property of *enhanced backward secrecy* is of class DRG.3, and a DRG.3 generator which verifies the property of *enhanced forward secrecy* is of class DRG.4.

> *Remark :* For a purely deterministic generator, by definition, the output of the generator is completely determined by the state of the system. The *enhanced forward secrecy* property can then only be verified if the seed is reinstantiated after each number generation.

**2.4.2. Deterministic generator architectures recommended by the SP 800-90A standard.** Considering the specific needs for uniformity of the generated data, and, depending on the desired level of security, the difficulty of inversion without knowledge of the seed, cryptographic functions appear to be good candidates for deterministic functions for randomness generation. In this regard, the NIST, through its SP 800-90A standard [**4**], offers a set of recommendations for algorithms for random number generation based on cryptographic functions.

More specifically, the recommended algorithms are the following: Hash_DRGB, HMAC_DRGB and CTR_DRGB.

**Specification of the** Hash_DRGB**:** The **seed** $V$ is **initialized** by concatenating a first output from the entropy source with a nonce and optionally additional data. The **first output** $C$ of the generator then consists of a double hash of the seed as described in Alg. 1, in which || represents concatenation and Hash_df is a data derivation function of arbitrary length (here $seedlen$) based on a hash function.

**Algorithm 1** Hash_DRGB: Computation of the first output $C$.

**Input:** entropy_input, nonce, seedlen > 0, additional_string. ▷ additional_string is optional.
**Output:** First output $C$.

$V = \texttt{Hash\_df}((entropy\_input||nonce||[additionnal\_string]), seedlen)$
$C = \texttt{Hash\_df}((0x00, V), seedlen)$
$reseed\_counter = 1$

**Seed reinstantiation** is performed in the same way as the initialization, by replacing the nonce with the previous seed $V$.

Finally, the **random data** $W$ is **generated** according to the procedure described in Alg. 2, with Hash being the hash function used by Hash_df.

**Algorithm 2** `Hash_DRGB`: Computation of the random data $W$.

**Input:** max_reseed_counter > 0, seedlen > 0, additional_string. ▷ additional_string is optional.
**Output:** Random data $W$.
  **if** reseed_counter > max_reseed_counter **then**
    reinstantiate the seed
  **end if**
  **if** additional_string $\neq null$ **then**
    $V = (V + \texttt{Hash}(0x02||V||additional\_string) \bmod 2^{seedlen}$
  **end if**
  $data = V, W = null$
  **for** $i = 1$ to $m$ **do**
    $w = \texttt{Hash}(data)$
    $W = W||w$
    $data = (data + 1) \bmod 2^{seedlen}$
  **end for**
  $V = (V + \texttt{Hash}(0x03||V) + C + reseed\_counter) \bmod 2^{seedlen}$
  reseed_counter = reseed_counter +1

We can see that, provided the nonce and the initial data of the entropy source are kept secret, this generator verifies the properties of forward secrecy and enhanced backward secrecy. Indeed, hash functions are so-called *one-way functions*, which means that it is easy to compute the images of the function, but it is not possible to predict with any significant advantage (in the probabilistic sense), a preimage for a given image. In particular, even with the knowledge of every variable of the algorithm at a given instant $t$, it is not possible to get back to preceding values of $V$, and then to the data $W$ previously generated (Cf. steps (3) and (4) of the generation procedure). This generator therefore satisfies the properties of forward secrecy and enhanced backward secrecy defined by the AIS 20/31.

However, as explained in subsection 2.4.1, to verify the enhanced forward secrecy property, the value $max\_reseed\_counter$ must be set to 1 so that the seed is reinstantiated after each data generation. Without reinstantiation of the seed, the set of values of the algorithm's variables may indeed be derived from the current values at a given instant.

> *Remark:* Although not specified in the standard, to guarantee a maximum level of security, the input data to the hash function must imperatively have an entropy greater than the length of the data it outputs. Indeed, the entropy of the output of a hash function (like any deterministic function) will always be at most equal to the entropy of the input.

The principle of the `HMAC_DRGB` is relatively similar, but a key-parameterized `HMAC` function is used instead of the simple hash function of the `HMAC_DRGB`.

Finally, in the case of the `CTR_DRGB` generator, a counter-mode block cipher algorithm (e.g. AES-CTR) is used, also parameterized by a key and optionally some additional data.

**2.4.3. Limits of deterministic random number generators.** DRNGs have the advantage of being easy to implement, even in hardware. However, their major limitation lies in the necessity of keeping secret all or part of the initial data, or in the constraint of a very frequent reinstantiation of the seed to guarantee a maximum level of security.

In addition, the security of these generators relies on the security of the cryptographic functions. By way of example, it is theoretically feasible to develop a generator based on Linear-feedback shift registers (abbreviated in **LFSR**), as developed by Durga et al [**24**]. LFSRs are stream encryption algorithms (which deliver encrypted bits of data continuously, by opposition to block encryption methods), and are, as such, potential candidates to build deterministic random number generators. Their security lies in the fact that their architecture must remain secret. It is known, however, that LFSRs are vulnerable to attacks using the Berlekamp-Massey algorithm [**10, 56**], which computes the LFSR of minimum size that has produced a given data set. If such an attack is successful, an LFSR-based generator no longer verifies the persistent confidentiality property, and is therefore completely compromised.

This example shows the importance of the choice of the generation function used for a DRNG, and the safety risks that may arise in the event that this choice is not made rigorously. It also showcases the importance of having standardized architectures that developers can rely on, such as the ones proposed in the SP 800-90A.

Finally, all DRNGs have a theoretical limitation, which is the actual existence of one-way functions, an open problem equivalent to the "$P$ versus $NP$" problem [**18**]. Indeed, based on [**11, 80, 49, 32**], the authors of [**39**] show that "the existence of one-way functions is necessary and sufficient for the existence of pseudorandom generators", meaning that if one-way functions do not actually exist, deterministic random number generators could not exist either. Although this limitation goes arguably way beyond the scope of random number generators, as the hypothesis that one-way functions exist is a requirement for the security of most cryptographic systems.

## 2.5. Non-deterministic random number generators

Despite their performance, we have seen that DRNGs require strong constraints (notably with regard to the secrecy of certain data) and guarantees on the security of the functions used to obtain a guarantee on the level of security of the generator as a whole.

In the case of non-deterministic generators, it is possible to achieve a high theoretical level of security without having to keep the internal state of the generator secret, since the phenomena used in the generation process are intrinsically random. This is particularly the case for PTRNGs, which are based on a physical phenomenon.

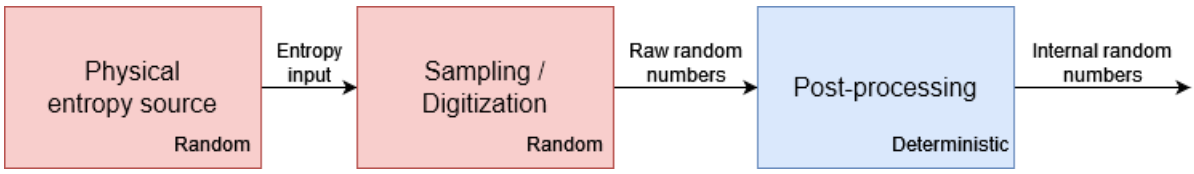



FIGURE 2. Schematic view of a generic PTRNG.

For any PTRNG, the overall principle is the same: an intrinsically random physical phenomenon, called the *entropy source*, is sampled to produce what is called by the AIS 20/31 standard a *raw random number* (see §1.4 of [**43**] or the glossary of [**67**]). These raw random numbers are then post-processed to correct for any statistical defect that may remain after the physical phenomenon has been sampled. These numbers are referred to as *internal random numbers*, and are ready to be delivered by the generator.

To fit our generic description of generators from subsection 2.3, the random part of a PTRNG is the physical phenomenon and the signal it produces, as well as the sampling and digitization of this signal,[3] and the deterministic part is the post-processing on the digitized data to produce random numbers.

Similarly to the classes for deterministic generators, the AIS 20/31 also provides a set of classes for PTRNGs: `PTG.1`, `PTG.2` and `PTG.3`. For the `PTG.1` class, the prerequisites are having a total entropy source failure test, which is triggered when the physical source no longer produces any entropy, and on-line tests on the internal random numbers (after post-processing) which verify that these numbers have good statistical properties. In addition, the internal numbers must pass a battery of black-box statistical tests, named *Procedure A*, which we describe in more details in subsection 3.2.5.1.

Class `PTG.2` has the prerequisites of class `PTG.1`, but this time the on-line tests are applied to the raw numbers, before any eventual post-processing. These on-line tests must also be chosen so as to precisely test the possible statistical defects of the raw numbers. To do this, the choice of tests must be based on a stochastic model of the entropy source (see point 290 on page 76 of [**43**]). In addition, the Shannon entropy must exceed 0.997 per bit. The *Procedure B* (described in subsection 3.2.5.2) is to be applied on raw random numbers, and can provide an estimation of the entropy per bit of the tested sequence. According to the authors of the standard, this class of generators can then be used for cryptographic applications such as key generation, random padding or even to provide seeds for deterministic generators, such as those introduced in subsection 2.4.2.

[3] In the draft of the AIS 20/31 ([**67**] §5.4.1, (929)), digitization mechanisms are indeed considered random "*due to (inadvertent) band-pass filtering, inherent noise, and probabilistic detection*" which may "*undesirably blur even a physically perfect noise signal or introduce dependencies between samples*".

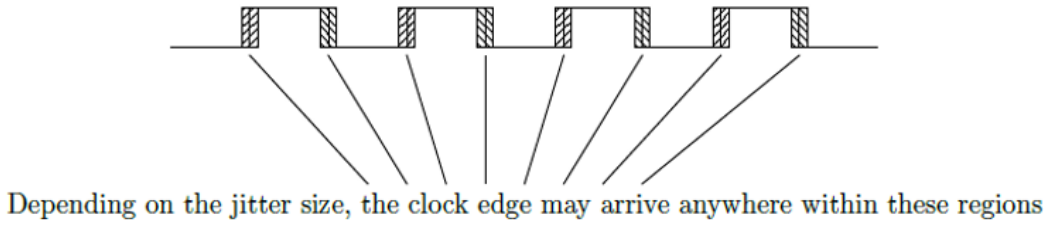



FIGURE 3. Clock jitter on a square-wave signal [**68**].

Finally, the prerequisites of class PTG.3 include all the prerequisites of class PTG.2, with the addition of the requirement for a cryptographic post-processing, more specifically the use of a class DRG.3 deterministic generator as post-processing. In other words, a class PTG.3 generator is a class PTG.2 physical generator that supplies seeds to a class DRG.3 deterministic generator.

> *Remark:* In the draft of the new version of AIS 20/31 [**67**], only classes PTG.2 and PTG.3 remain, the authors having considered that class PTG.1 was of no practical use. Class PTG.2 has also been made stricter, as the Shannon entropy of internal numbers must now exceed 0.9998 per bit. A new entropy measure, the minimum entropy or min-entropy, is also accepted, and its value must exceed 0.98 per bit.

The ANSSI, through its guide PG-083 [**1**], does not offer a classification of generators based on security levels, as presented in AIS 20/31, but does propose a number of rules and recommendations for the design of generators, based on the premise that it is difficult today to justify the quality of a physical source of randomness, but easy to prove the robustness of a post-processing algorithm.

In particular, ANSSI *stipulates* that the physical generator must have a "functional description" to justify it generates "true randomness", and that a cryptographic post-processing with an internal state (a set of secret data such as keys) of at least 128 bits must be used. In this sense, the PG-083 only authorizes physical generators equivalent to class PTG.3 of AIS 20/31. The guide further *recommends* that the internal state of the cryptographic post-processing has 256 bits.

**2.5.1. Ring oscillator-based random number generators.** Any partially random physical phenomenon that can sampled can be used to generate random numbers. In practice, the most commonly used ones are phenomenon of an electronic nature, such as *clock jitter* [**34, 35**], due, among other things, to the ease of implementing generators based on these phenomena in processors. Clock jitter is the uncertainty that exists on the precise timing of the rising or falling edges of an oscillating signal (see Fig. 3).

One way of taking advantage of clock jitter is to use ring oscillators, which are a sequence of odd-numbered inverters (logic NOT gates) used to create a raw oscillating signal that can be

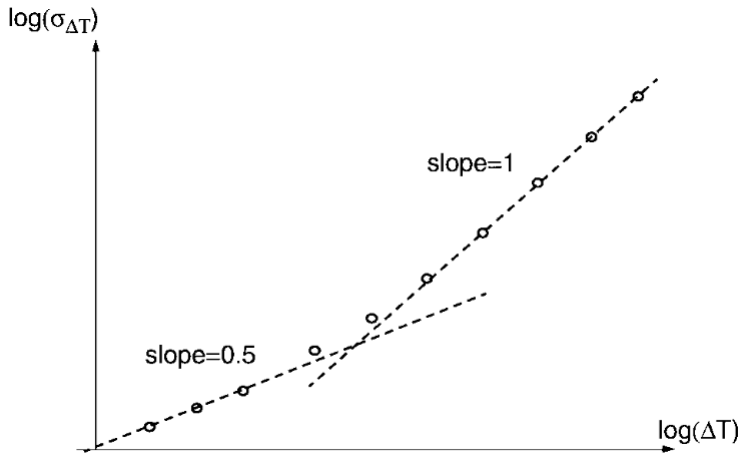



FIGURE 4. Evolution of the clock jitter amplitude as a function of the signal sampling period, *log-log* scale [**35**].

used as a clock. In an theoretical setting, the frequency of the clock signal is perfectly constant, and in particular, the timings of the rising and falling edges of the signal are completely determined by the number of inverters in the oscillator. In practice, however, the signals from ring oscillators are affected by jitter, which causes uncertainty in these clock edge timings.

In [**35**], Hajimiri *et al.* explain that clock jitter becomes increasingly important as the number of periods between two sampling operations increases (this number of periods is also known as the *accumulation time*). The authors also explain that this clock jitter originates from two categories of electronic noise: uncorrelated noise such as thermal noise [**62**] and correlated noise such as power supply noise or flicker noise [**37**]. Uncorrelated noises are predominantly present at low accumulation times and cause a linear increase in the jitter's variance, while correlated noises tend to appear at high accumulation times and cause a linear increase in the jitter's standard deviation as a function of sampling period (See Fig. 4). As the name suggests, this correlated noise introduces correlation between the jitters affecting successive edges of the clock signal. From the point of view of random number generation, the aim is to find the right compromise between an accumulation time high enough to obtain a suitable amount of jitter, but low enough to keep the proportion of uncorrelated noise higher than that of correlated noise.

In order to take advantage of jitter to produce random numbers, several TRNG constructions based on ring oscillators (abbreviated RO-TRNG) have been proposed, including the Elementary RO-TRNG (ERO-TRNG) and the Multiple rings RO-TRNG (MURO-TRNG), which we will describe in detail below. Jitter-based generators aim at sampling an oscillating signal when it is impacted by jitter (i.e. on a rising or falling edge timing) to produce random data.

2.5.1.1. *ERO-TRNG.* The Elementary RO-TRNG (ERO-TRNG) is based on two ring oscillators placed at the input of a flip-flop. One oscillator serves as a data generator, and the other as a sampling clock. A frequency divider is often placed in front of the sampling oscillator to increase the accumulation time. As both oscillators are affected by jitter, the overall system is also impacted by a relative jitter, representative of the quality of the randomness supplied by the generator. The greater the relative jitter, the greater the uncertainty on the sampled values, and therefore the better the quality of the randomness. The architecture of the ERO-TRNG is illustrated in figure 5.

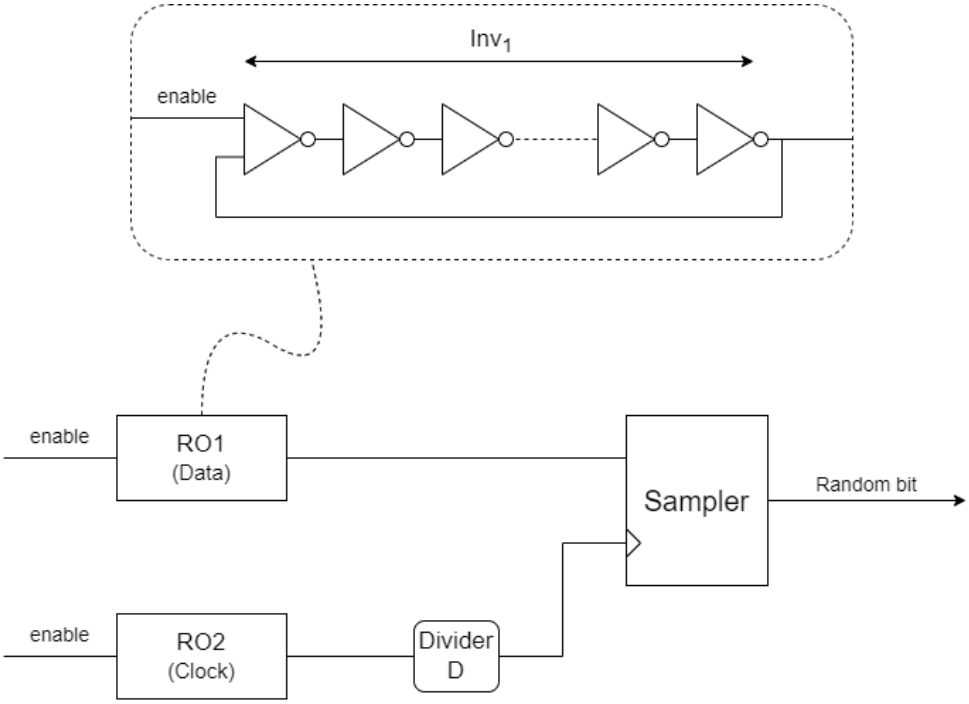



FIGURE 5. Schematic representation of the ERO-TRNG.

2.5.1.2. *MURO-TRNG.* The Multiple ring RO-TRNG (MURO-TRNG) is based on the same principle as the ERO-TRNG, but looks to palliate a problem of the latter, which is that the accumulation time must be relatively long to produce randomness of sufficient quality. To reduce the accumulation time needed to obtain a satisfactory randomness, several oscillators are placed in parallel and their outputs are placed at the input of an XOR gate to serve as the data signal at the input of the sampling flip-flop. As each data oscillator possesses its own jitter independently of the others, the relative jitter at the output of the XOR gate corresponds to the sum of the $n$ jitters, which means that the accumulation time can be divided by the same factor $n$. On the other hand, the drawback of this architecture is its increased need for electric power and surface on the circuit board.

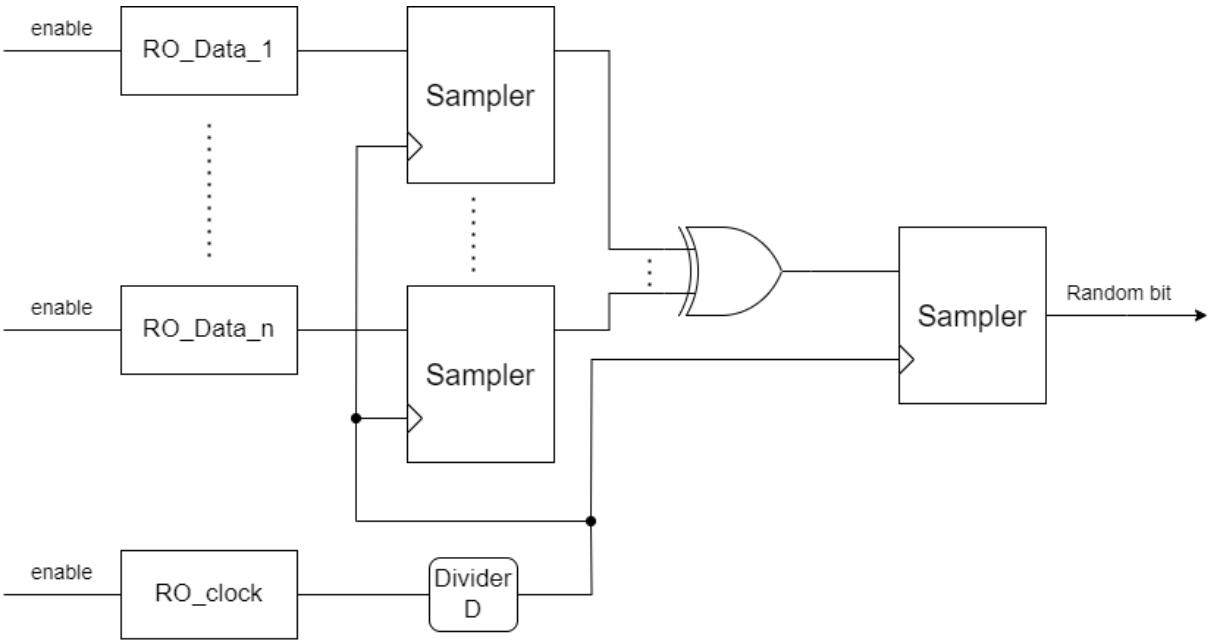



FIGURE 6. Schematic representation of the MURO-TRNG.

2.5.1.3. *TERO-TRNG.* Finally, not all random number generators based on ring oscillators use clock jitter as a random phenomenon. For example, the Transient Effect RO-TRNG (TERO-TRNG) is based on the phenomenon of metastability. More specifically, in this generator, two ring oscillators are placed in a loop and activated synchronously. Such a construction (described in Fig. 7) leads to the so called metastability effect, which means that the signal produced is unstable for a short time after activation of the two oscillators, before stabilizing. In order to produce random numbers, the principle is therefore to sample the signal during its unstable (and therefore unpredictable) phase.

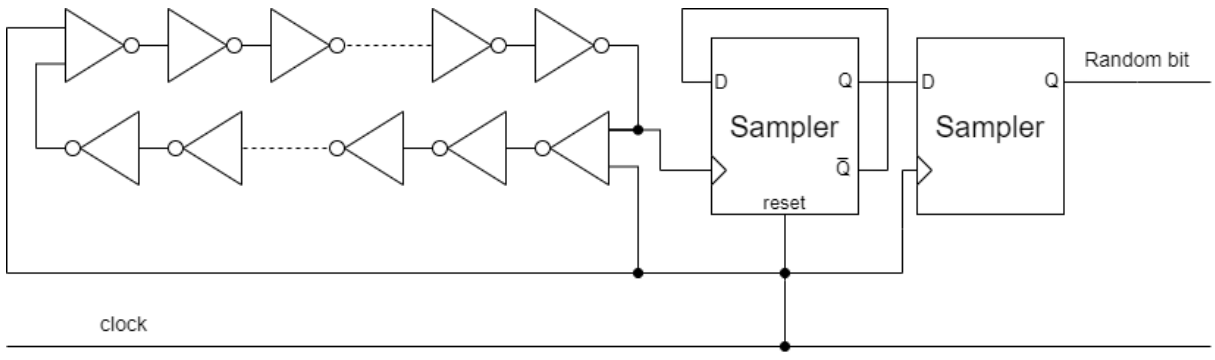



FIGURE 7. Schematic representation of the TERO-TRNG.

**2.5.2. PLL-based random number generators.** Random number generators based on *Phase-Locked Loops* (PLL) also take advantage of the clock jitter to produce random numbers. The principle of a phase-locked loop is as follows: a base clock signal is placed at the input of a phase comparator, which feeds a voltage-controlled oscillator (whose frequency is controlled by the input voltage). The oscillator's output is then reinjected to the phase comparator's input to form a feedback loop that attempts to maintain the oscillating signal at the frequency of the initial clock signal. The principle is illustrated in figure 8 [**28**].

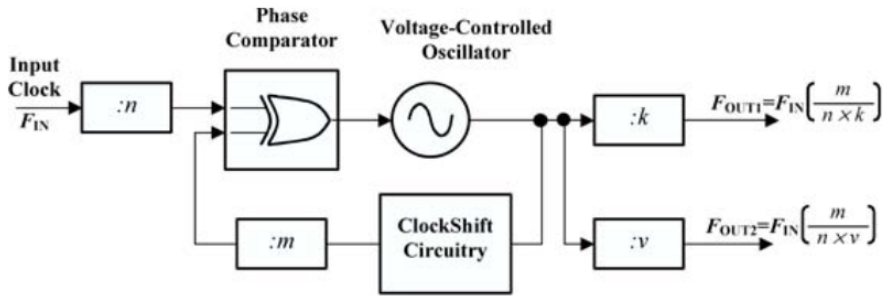



FIGURE 8. Block diagram of a PLL circuit on Altera. [**28**].

Very similarly to the ring oscillators described above, the voltage-controlled oscillator is subject to clock jitter, which disturbs its oscillation frequency. The idea is to accumulate a certain number of samples of this jitter-disturbed signal to produce random data. Figure 9 shows one of the constructions used to generate random numbers in this way. The data are sampled by a $D$ flip-flop, and each set of $K_D$ data is passed through a XOR decimator, which sums the data in pairs (with an *exclusive-or* operation) to produce a single random bit.

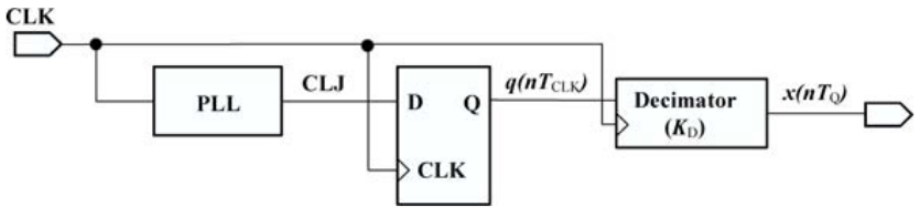



FIGURE 9. Randomness extraction from a PLL. [**28**].

This very simple architecture is similar to the ERO-TRNG, and can be made more complex by, for example, placing several PLLs in parallel in a manner analogous to the MURO-TRNG architecture, in order to increase the data generation throughput (Cf. [**28**] §3.3).

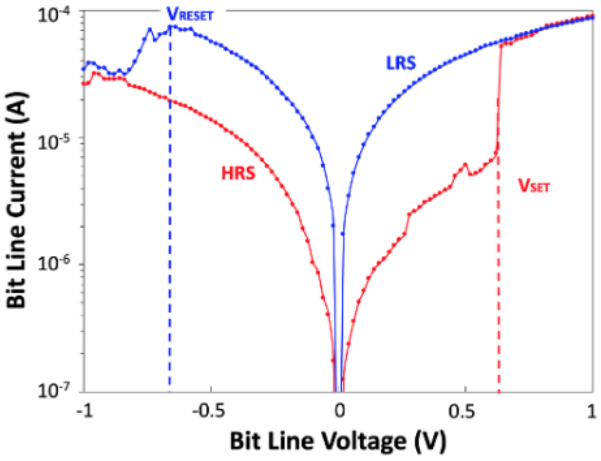



FIGURE 10. Links between voltage applied across electrodes and cell resistivity. [**3**].

**2.5.3. Generators based on memory cells.** Although generators based electronic phenomena are widely used due to their ease of integration into existing circuits, other phenomena can allow for the generation of random numbers. Memory cell oxidation phenomena, for example, can be used to generate Physical Unclonable Functions (PUF), which serve as unique identifiers for physical systems, but can also be derived into random number generators.

The authors of [**3**] and [**66**] propose random number generator architectures based on RRAM (Resistive Random Access Memory) cells, which encode binary information according to the electronic resistance of each cell. For example, a 'high resistance state' (HRS) encodes a 0 and a 'low resistance state' (LRS) encodes a 1 . In the case of the two articles, the memory cells are more precisely of the OxRAM (Oxide-based RAM) type, whose principle is as follows: memory cells consist of two metal electrodes separated by a dielectric layer. Normally, these cells are insulating. However, when a sufficiently large potential difference is applied across the electrodes, a conductive filament forms (SET step) between the two electrodes (in the case of both articles, an oxygen cavity filament), and the cells become conductive. This phenomenon can then be reversed by applying a negative potential difference to reabsorb the conductive filament (RESET step). This cycle is summarized in Fig. 10.

In the above graph, the terms LRS and HRS denote low and high resistance states respectively, and the terms $V_{SET}$ and $V_{RESET}$ denote SET and RESET voltages respectively.

However, these voltages, and in particular the SET voltage, are also variable for each cell and depend, among other things, on the state of the dielectric layer, which changes with each SET and RESET operation. To produce random data, the idea is to apply an identical voltage across

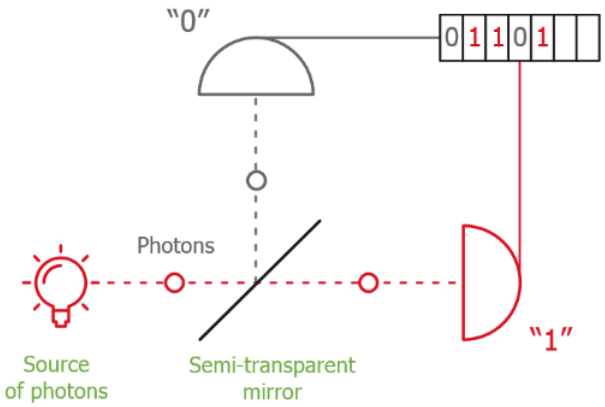



FIGURE 11. Schematic representation of a QRNG based on a semi-reflective mirror. [**71**].

the electrodes, so that around 50% of the cells are oxidized (appearance of the conductive filament). In this case, the generator produces a random matrix after each SET step, containing around 50% of bits 1. However, the generator presented in [**3**] suffers from data dependencies before post-processing (see Table II on page 7 in their article).

**2.5.4. Generators based on quantum phenomena.** In addition to electronic and electrochemical phenomena, one of the most intrinsically random phenomenon is the quantum behavior of elementary particles (especially photons). This is sometimes referred to as QRNG, for *Quantum Random Number Generator*. One way of generating random numbers from photons (Cf. [**71**] §4), for example, is to use a laser to project photons onto a semi-reflective mirror, behind which two sensors are placed (one facing the laser to capture unreflected photons, the other at 90° to capture reflected photons, as shown in Fig. 11). One of the photonic sensors is then used to encode a bit 0, while the other encodes a bit 1.

In the case where the mirror is perfectly semi-reflective, the proportion of bits 0 and 1 is perfectly balanced. However, it is very difficult in practice to obtain a reflection rate of exactly 50%. For example, for the Quantis generator from the ID Quantique company, the authors of the white paper [**71**] simply announce a bias of less than 10%, meaning that the proportion of bits 0 (and conversely of bits 1) lies between 45% and 55% before post-processing.

A second possibility for generating random numbers from photons is to use another random property of photons, which is that, for a light source, the number of photons emitted during a given period of time is non-deterministic (more precisely, it follows a Poisson distribution [**76**]). Another generator architecture proposed by ID Quantique is based on a photo-diode emitting photons onto a sensor array. The number of photons detected by each sensor thus produces a random number. The authors of [**71**] (§4.2) then claim that the output numbers

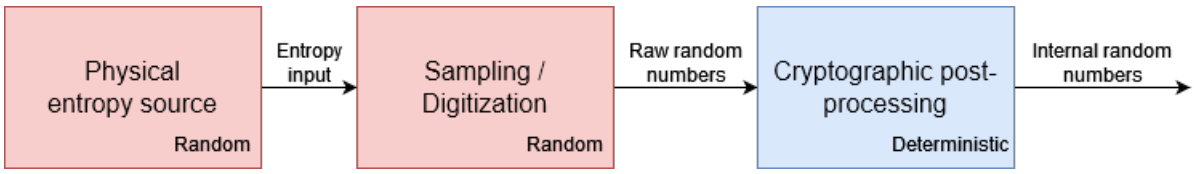



FIGURE 12. Schematic view of an Hybrid RNG.

of this generator already have maximal entropy, hence a perfectly random behavior, without even the need for post-processing. To back up their claim this, they rely on a set of standard tests, some of which will be discussed in the next section.

**2.5.5. Limits of true random number generators.** In this section, we have thus seen that the means of generating random numbers are extremely diverse, due to the multiplicity of physical phenomena that can be used, and the different possible architectures taking advantage of a given phenomenon. However, these generators are not flawless either. The fact that they rely on a physical phenomenon make them subject to the environmental conditions, or aging, which deteriorate the quality of the phenomenon in terms of the randomness it leads to, up to a total loss of entropy. The physical entropy source is also a target of choice for attackers, as it opens up a larger variety of vulnerabilities that can be exploited to reduce the quality of the randomness produced by the generator. We will go in more details on attacks on TRNGs in Chapter 8. To limit the impact of a deterioration of the physical entropy source (whether voluntary or not), choosing a good post-processing is then a good solution. The post-processing of a TRNG can even be a DRNG in and of itself, with such generators being called *Hybrid Random Number Generators*.

## 2.6. Hybrid Random Number Generators

As aforementioned, Hybrid Random Number Generators are the combination of a PTRNG and a DRNG (more precisely, a DRNG based on a cryptographic function, equivalent to class DRG.3 proposed by the AIS 20/31 [**67**] (192)). The DRNG then serves as a **cryptographic post-processing** to the raw numbers produced by the PTRNG. It ensures that, should the physical entropy source degrade over time, no statistical anomaly will be seen at the output of the generator.

However, as presented in the beginning of Sect. 2.5, the PTRNG part of an Hybrid Random Number Generator is still subject to online tests, which raise an alarm when the statistical properties of the raw random numbers degrade beyond an acceptable point. The security of an Hybrid RNG should therefore never be equal to the sole security of its DRNG part. More precisely, by contrast to simple TRNGs or DRNGs, the security of an hybrid RNG relies on both the information-theoretic properties of the TRNG part (provided by the stochastic model), and the computational security of the DRNG part (provided by the analysis of the underlying cryptographic function). Hybrid RNGs then theoretically offer the highest security level possible, and, under these circumstances, shall be used for cryptographic applications.

## 2.7. Conclusion

In practice, for these different generators (deterministic, non-deterministic, or hybrid) to be used in a cryptographic context, it is necessary to be able to offer guarantees on the quality of the randomness they produce. These guarantees can be provided by means of a precise modeling of the generator, or by statistical measurements on its output for example. The aim of the following chapter will therefore be to study the different ways of ensuring that a generator meets a certain level of quality, depending on the type of generator encountered.

CHAPTER 3

# Evaluating the quality of the randomness

No matter the type of evaluation carried out on a cryptographic system, the goal is always to ensure that an attacker will not be able to gain an advantage on the evaluated system to compromise its security. In the case of a random number generator, gaining an advantage would mean being capable of anticipating, even partially, the sequence of numbers produced by the generator. More precisely, regardless of the type of generator (physical or not, deterministic or not), the produced numbers must be indistinguishable from ideal random numbers because, as we saw in the introduction, even partial knowledge of the generated numbers can compromise the security of a cryptographic scheme.

As a matter of example, if strong correlations exist between bits of successive nonces in an ECDSA scheme (meaning that some bits of future nonces can be estimated with a non-zero probability with the knowledge of previously generated nonces), this could lay the groundwork for an attack such as the one presented by Nguyen and Shparlinski [**60, 61**], or some other variation of this attack.

It therefore appears essential to have robust methods and metrics to ensure that the generators used produce an randomness of optimum quality.

In the PG-083 guide, the ANSSI does not impose a strict roadmap on the methods of assessment of the quality of the randomness produced by a generator. In the following paragraphs, the quoted sentences are our proposed translations for the original sentences in French in the PG-083 guide. See the different footnotes for the original phrasings.

As far as statistical tests are concerned, it simply stipulates that "*statistical tests on the output of the generator must not reveal any significant defect in the generated random numbers*".[1] The authors mention the FIPS 140-2 and SP 800-22 tests; the former ones are also included in the AIS 20/31 test battery, detailed later in this chapter. But they also explain that "*any test that appears to be relevant can be used*"[2] on the sequences before any post-processing.

[1] "Des tests statistiques en sortie du générateur physique ne doivent pas faire apparaître de défauts significatifs dans l'aléa généré.", [**1**] §2.4.2, RègleArchiGVA.2.

[2] "mais tout test paraissant pertinent peut être utilisé.", [**1**] §2.4.2, last point in *Justification.*

On the post-processing, the authors of PG-083 state that the cryptographic primitive used must be "*compliant with the referential*".[3]

The guide's authors also mention that choices of physical source and architecture in particular must be justified by reasoning, whether "*heuristic or rigorous, qualitative or quantitative*".[4] Here again, designers are relatively free in their approach of the proof, the key point being to convince that "*the generator does indeed produce true randomness*".[5].

In the case of AIS, the evaluation of random number generators is much more guided, once again through the different safety classes. In particular, the most restrictive class on physical generators (PTG.3) requires that a stochastic model of the physical entropy source must be provided to prove its quality. In addition, a specific statistical testing procedure must be applied to further validate the quality of the randomness produced by the generator.
In this chapter, we will then detail these two approaches, using stochastic modeling and statistical testing, to explain their use cases and interest, as well as their limitations.

## 3.1. Entropy measurement

**3.1.1. The notion of entropy.** In [**48**], Leinster describes the Shannon entropy [**73**] as a measure of the "information" or equivalently of the "expected surpise" gained from the observation of random events (see p.40 in [**48**]). As a reminder, the Shannon entropy of a probability distribution is defined as follows:

DEFINITION 3.1 (Shannon entropy [**73**]). *For $\boldsymbol{p} = (p_1, ..., p_n)$, $n \in \mathbb{N}$ a probability distribution on n random events, the Shannon entropy H of $\boldsymbol{p}$ is equal to:*

$$H(\boldsymbol{p}) = -\sum_{i=1}^{n} p_i \log(p_i).$$

> *Remark:* Some probabilities in $\boldsymbol{p}$ can be null if we accept the convention $0 \times \log(0) = 0$. Also, the base of the logarithm is not specified in the definition as it only adds a constant multiplier to the value of the entropy. In the context of information theory, the base of the logarithm is often chosen to be 2.

We know that, when $p_i \to 1$, $\log p_i \to 0$, and when $p_i \to 0$, $p_i \times \log p_i \to 0$ as well. Intuitively, in such cases, an observer will not be surprised to see (or not see) the event $j$ occur, and $H(\boldsymbol{p})$ can be seen as the expected surprise of the observation of the events described by the probability distribution $\boldsymbol{p}$.

[3] "Les primitives cryptographiques employées par le retraitement algorithmique doivent être conformes au référentiel.", §2.4.3, RègleAlgoGDA.1.

[4] "justifier les choix faits par un raisonnement, qu'il soit heuristique ou rigoureux, qualitatif ou quantifié.", [**1**] §2.4.2, third point in *Justification*.

[5] "La forme et le type de raisonnement sont laissés libres. Son but est de convaincre [...] que le générateur d'aléa produit bien de l'aléa vrai", [**1**] §2.4.2, third point in *Justification*.

Alternatively, Leinster also presents the use the Shannon entropy in the description of ecological communities, and more precisely in the description of their distribution. In sect. 2.4 of his book, Leinster presents the notion of biological *diversity*, especially used in Biology, and which serves to describe the relative abundance distribution of different species in a community. Intuitively, it is a measure of how evenly the different species are balanced in the community, and it is formally defined as follows:

DEFINITION 3.2 (Diversity of order 1). *For $\boldsymbol{p} = (p_1, ..., p_n)$ a distribution on n events, the diversity of order 1 of $\boldsymbol{p}$, denoted by $D(\boldsymbol{p})$ is the geometric mean of $\boldsymbol{p}$, i.e.:*

$$D(\boldsymbol{p}) = \left(\frac{1}{p_1}\right)^{p_1} ... \left(\frac{1}{p_n}\right)^{p_n} = \frac{1}{p_1^{p_1} ... p_n^{p_n}}.$$

> *Remarks:* Again, with the convention that $0^0 = 1$, some probabilities in $\boldsymbol{p}$ can be null. Also, the terms "of order 1" relate to the average function used, here the geometric mean. Other functions can be used, but we will not mention them in this manuscript. We redirect the interested reader to [**48**] (§4.2) for the description of other mean functions, named *power means*.

We then notice that $D(\boldsymbol{p}) = \exp(\ln(\eta) \times H(\boldsymbol{p}))$, where $\eta$ is the base of the logarithm chosen for the entropy. In particular, when $\eta = 2$, we have $D(\boldsymbol{p}) = 2^{H(\boldsymbol{p})}$. The notion of entropy is thus completely equivalent to the notion of diversity.

The property of the Shannon entropy which makes it very interesting when studying the diversity of species is that it is null when one of the probabilities is equal to 1 (and the other are then all equal to 0), and it is maximal when all of the probabilities are equal to $1/n$, so when all of the events (or species) are evenly distributed. More specifically, when all events are equally likely, $H(\boldsymbol{p}) = \log(n)$. See Lemma 2.2.4 in [**48**] for the proof.

In the context of random numbers, the notion of diversity can be directly applied to the distribution of $k$-bit numbers for example, each number representing a species. For an ideal generator, each $k$-bit number is equally likely to appear, which mean that the diversity, or entropy of the distribution of numbers is maximal. Thus, when measuring the entropy of the distribution of $k$-bit numbers, any value significantly lower than $k$ will be revealing of a disproportion in the numbers, which makes the output of the generator more predictable.

While the Shannon entropy is the most wide-spread, it is not the only entropy available. In particular, a generalization of the Shannon entropy, called the Rényi entropy exists and is defined as follows:

DEFINITION 3.3 (Rényi entropy of parameter $\alpha$ [**69**]). *For a real number $\alpha \in \mathbb{R} \setminus \{1\}$, $\alpha \geqslant 0$, and a probability distribution $\boldsymbol{p} = (p_1, ..., p_n)$, $n \in \mathbb{N}$, the Rényi entropy of parameter $\alpha$ of $\boldsymbol{p}$ is equal to:*

$$H_\alpha(\boldsymbol{p}) = \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n} p_i^\alpha\right).$$

This generic entropy allows for the definition of another specific entropy which is commonly used alongside the Shannon entropy in the evaluation of random number generators : the min-entropy.

DEFINITION 3.4 (Min-entropy). *For a probability distribution $\boldsymbol{p} = (p_1, ..., p_n)$, $n \in \mathbb{N}$, the min-entropy of $\boldsymbol{p}$ is equal to:*

$$H_{min}(\boldsymbol{p}) = \min_{1\leqslant i\leqslant n} -\log\left(p_i\right) = -\log\left(\max_{1\leqslant i\leqslant n}\left(p_i\right)\right).$$

In the context of random number generation, this entropy can be seen as a measure of the maximal deviation of the distribution of numbers to the ideally random model. As such, it gives a lower bound on the "quality" of the randomness produced by a generator.

Often, when evaluating random number generators, the value of interest is the *entropy per bit*, defined as follows:

DEFINITION 3.5 (Entropy per bit of a probability distribution). *For $n \in \mathbb{N}$, $n > 0$, and $\boldsymbol{p} = (p_1, ..., p_n)$ a probability distribution on n events, the entropy per bit of $\boldsymbol{p}$ is equal to:*

$$\frac{H(\boldsymbol{p})}{\log_\eta(n)},$$

*where $\eta$ is the base of the logarithm used to compute the entropy.*

For example, in the case that $n = 2^k$, $k \in \mathbb{N}$, the probability distribution $\boldsymbol{p}$ can be seen as a distribution on the $k$-bit words. And, for $\eta = 2$, the entropy per bit is expressed as $H(\boldsymbol{p})/k$.

The entropy per bit allows for a standardization of the value of the entropy at the output of different generators, independently from the size of the output space of each generator. This value can then be used to define a common threshold that every generator should aim to attain, as we will see for example in the following subsection.

> *Note:* In cryptography, the notion of entropy (especially the Shannon entropy) is often used as a metric of robustness of crypto-systems against attacks such as the exhaustive search (also called brute force) of a secrete key. As an example, in the case of secrete key of 3 bits $(b_1, b_2, b_3)$, represented by the variables $(B_1, B_2, B_3)$, when the 3 bits are uniformly random and independent, the Shannon entropy of the distribution of $(B_1, B_2, B_3)$ is equal to 3, and an exhaustive search will take on average $2^{3-1} = 4$ tries to find the correct key. However, if for example $B_2$ and $B_3$ are constrained by one another such that, $b_2 = b_3$, then the distribution $p'$ of $(B_1, B_2, B_3)$ is:

$$p' = (\Pr(0,0,0), \Pr(1,0,0), \Pr(0,1,0), \Pr(1,1,0), \Pr(0,0,1), \Pr(1,0,1), \Pr(0,1,1), \Pr(1,1,1)),$$
$$= \left(\frac{1}{4}, \frac{1}{4}, 0, 0, 0, 0, \frac{1}{4}, \frac{1}{4}\right),$$

and the entropy of $p'$ is equal to:

$$H(p') = -4 \times \frac{1}{4} \times \log_2\left(\frac{1}{4}\right) = 2.$$

In such a case, an exhaustive search would only need to guess the first two bits, as the bit $b_3$ is deduced from $b_2$. On average, an exhaustive search then requires $2 = 2^{H(p')-1}$ tries to guess these two bits.

While it is very specific, this example showcases how the entropy can be used to estimate the robustness of systems against certain attacks (although the link between the value of the Shannon entropy and the complexity brute force attacks is further discussed in [**55**], and appears less direct for more general distributions).

**3.1.2. Stochastic modeling and entropy measurement.** The stochastic model of a random number generator is a probabilistic description of the outputs of the generator, as a function of its parameters. When evaluating a physical generator using a stochastic model, the aim is to translate the parameters of the model into a measure of the entropy. Since the stochastic model is based on the physical model of the phenomenon used, entropy must be derived by estimating the parameters of this physical model. This is, for example, what Lubicz *et al.* [**7, 29**] have achieved for the so-called "elementary" physical generator based on ring oscillators, or ERO-TRNG (see subsection 2.5.1 and Fig. 5).

The authors have modeled the behavior of oscillating signals from $RO_1$ and $RO_2$ as a function of various "fixed" generator parameters and the relative jitter of the two signals to formulate the following expression for the min-entropy per bit of the generator's output:

$$H_{min} = 1 - \frac{4}{\pi^2 \ln(2)} \times \exp\left(-\frac{D \cdot 4\pi^2 \sigma_{jit}^2 T_2}{T_1^3}\right),$$

where $H_{min}$ is the *min-entropy*, $T_1$ and $T_2$ are the periods of the oscillators $RO_1$ and $RO_2$, $D$ is the frequency division factor of the clock signal (from $RO_2$), and $\sigma_{jit}$ is the amplitude of the relative clock jitter between the signals from the two oscillators.

The developers of random number generators who can provide a stochastic modeling of their generator can then also provide a guarantee on the quality of the randomness it produces, as well as the set of parameters needed to get to the desired level of entropy. As a reference, in the AIS 20/31 [**43**], the required Shannon entropy per bit at the output of a generator is 0.997, and in the new draft [**67**], the Shannon entropy threshold has been raised to 0.9998, and a

min-entropy threshold has been placed at 0.98.

Having an expression of the entropy as a function of the parameters of the generator and of the physical phenomenon used also enables developers to precisely anticipate the impact of a given parameter on the quality of the random output. Here for example, we can verify that the amount of jitter $\sigma_{jit}$ is important to increase the entropy (which is expected due to it being the entropy source of the generator), but also that the larger the accumulation time, the better the quality of the randomness (the accumulation time being represented by both $D$ and $T_2$).

**3.1.3. Entropy estimation based on the generated data.** While using the stochastic model of a TRNG seems to be a very robust way to derive the value of the entropy of the source, other methods exist to estimate this entropy on the data produced by the generator. Such methods can be particularly useful when the model of the TRNG is not available.

In its standard SP 800-90B [**6**] (§6.3), the NIST offers a list of algorithms which can be used as estimators the min-entropy of a generator using the data it provides. However, these estimations are only correct up to a certain degree, with some algorithms providing only an upper or lower bound on the real min-entropy of the source (especially the compression estimate §6.3.4). Moreover, in [**33**], Hagerty and Draper state that the general problem of estimating the min-entropy of a random source based on a dataset is a constrained optimization problem, in which one must find the probability distribution which minimizes (or maximizes) the min-entropy based on the statistic that is used for the estimation. And this optimization problem is in all generality expected to be computationally difficult.

In this regard, while the estimators proposed by the NIST can serve as an indicator of the general quality of a generator, the entropy estimates they provide should only be viewed as indicators, and shall by no means constitute a definite proof of security for the RNG.

## 3.2. Statistical tests

As mentioned in the introduction, the stochastic model of a generator is not always available, either because the evaluated generator is not based on a physical phenomenon, or because the phenomenon has not yet been modeled. In this case, the application of black-box statistical tests (without taking advantage of any knowledge about the generator) is necessary.

> *Remark :* The existence of a stochastic model does not exclude the use of statistical tests. For example, even in the case of the most robust class of generators (`PTG.3`) for which a stochastic model is required, the authors of [**43**] require the application of statistical tests as a complement.

**3.2.1. Definition of a statistical test.** Statistical tests are formally defined as decision procedures, which aim at deciding whether to accept or reject an hypothesis "*based on the value of a certain random variable* $X$" (see [**47**], §3.1), or more specifically, based on the distribution $P_\theta$ of said variable $X$ (with $\theta$ the label of the distribution).

The hypothesis of the test will often be referred to as the *null hypothesis*.

The set of all possible distributions of $X$ (named $\mathscr{P}$) can then be partitioned into two classes of distributions, $H$ and $K$, meaning that $H \cup K = \mathscr{P}$ and $H \cap K = \emptyset$. $H$ corresponds to the distributions of $X$ for which the hypothesis is accepted, and $K$ to those for which the hypothesis is rejected. In practice, however, the distribution of $X$ is inaccessible, and one must refer to samples $x$ to decide on whether to accept or reject the hypothesis, or equivalently, on whether or not the distribution of $X$ can be in $H$. The set of all possible values $x$ of $X$ is then divided in two "complementary regions" $S_0$ and $S_1$, with $S_0$ the set of values $x$ for which the hypothesis is accepted, and $S_1$ the set of values for which it is rejected, also called the "*critical region*".

The random variable $X$ is called the *test statistic*, a statistic being defined as follows:

DEFINITION 3.6 (Statistic). *For $(\mathscr{X},\mathscr{A})$ a sample space, i.e. a measurable space (see Def. 2.2) based on a set of observable values, a statistic is a measurable transformation $T$ from $(\mathscr{X},\mathscr{A})$ into a measurable space $(\mathscr{T},\mathscr{B})$.*

In other words, a statistic $T$ is a random variable which transforms observable values into other observable values, and has for distribution $Q^T$ such that, for any event $B \in \mathscr{B}$:

$$Q^T(B) = P(T^{-1}(B)),$$

where $P$ is a probability measure over the sample space $(\mathscr{X},\mathscr{A})$.

In the context of statistical tests applied to binary sequences, this formal definition can then be summarized as such:

DEFINITION 3.7 (Statistical test on binary sequences). *For $V$ an arbitrary numerical set, $X : \{0,1\}^N \to V$ a random variable (the statistic of the test), and $S_0$ a subset of $V$, a statistical test on a binary sequence $b = (b_i)_{1\leqslant i\leqslant N}$ based on $X$ is a boolean function $T_{X,S_0}$, which returns a decision depending on whether or not the value of the statistic applied to the sequence belongs to the subset $S_0 \subset V$:*

$$\begin{array}{rccc} T_{X,S_0}: & \{0,1\}^N & \longrightarrow & \{true, false\} \\ & b & \longmapsto & X(b) \in S_0 \end{array}.$$

*If $T_{X,S_0}(b) = true$, the hypothesis is accepted and the test is* passed*, otherwise, the hypothesis is rejected the test is* failed.

**3.2.2. Type I and type II errors.** This type of test is associated with *type I* and *type II errors*, with respective probabilities $\alpha$ and $\beta$. These errors describe the case where the hypothesis has been incorrectly rejected (type I), or incorrectly accepted (type II). Their significance is summarized in Table 1, a "valid sequence" being a sequence which satisfies the statistical

| | **Valid sequence** | **Non-valid sequence** |
|---|---|---|
| **Test passed** | $1-\alpha$ | $\beta$ (Type II error) |
| **Test failed** | $\alpha$ (Type I error) | $1-\beta$ |

TABLE 1. Type I and II errors in hypothesis tests

hypothesis.

The terms $\alpha$ and $\beta$ are defined such that:

$$\mathrm{P}_\theta(X \in S_1) \leqslant \alpha, \text{ for all } \theta \text{ such that } \mathrm{P}_\theta \in H,$$

and

$$\mathrm{P}_\theta(X \in S_0) \leqslant \beta, \text{ for all } \theta \text{ such that } \mathrm{P}_\theta \in K,$$

with $H$, $K$ the classes of distributions defined in the previous subsection, $\mathrm{P}_\theta$ the distribution of $X$, and $S_1 = V \setminus S_0$.

These two errors and their parameters $\alpha$ and $\beta$ (the latter commonly referred to as the *power* of the test) are then used to define the set $S_0$ of "acceptable" values for the test statistic.

**3.2.3. Implementation and current standards on statistical tests.** In practice, and namely in the context of random number generators, statistical tests are most often designed to confront a dataset to the hypothesis that it fits to a given distribution.

Statistical tests specifically designed for random number generators exist at least since Kendall and Babington-Smith [**42**], who proposed a notion of *local randomness* (see [**42**], (21) in p.153), along with four tests on sequences of digits (numbers in $\{0,...,9\}$) aiming to test the hypothesis of a locally random sequence. The local randomness is the notion that, in a finite sequence of data, no digit should appear significantly more often than others, no pair of digits should be preponderant either, and so on for any collection of digits of any size in the sequence.

> *Remark:* The authors talk about the *local randomness* to describe the behavior that random numbers are expected to have in a finite sequence, by opposition to the idea that any set of digits, no matter how regular, can be viewed as a random sampling in an infinite sequence.

Other statistical tests applied to random number generators were later listed by Knuth in 1969 ([**44**] §3.3), and some are still commonly used today, sometimes slightly modified, such as the equidistribution test or the runs test (which will be detailed later).

More tests were subsequently carried out in batteries, such as the DieHard battery proposed by Marsaglia [**53**], which was later completed by Brown into the test battery DieHarder [**14, 15**]. In addition to these are the TestU01 by L'Ecuyer and Simard [**46**], and, perhaps most importantly, Maurer's universal test [**57**] (later reinforced by Coron and Nacache [**20**]), which provides an estimate of the entropy per bit of any binary information source.

Today, the design and evaluation of random number generators has been standardized, notably by the NIST [**4, 6, 5**] in the USA and the BSI [**43, 67**] in Germany, both of which offer a test battery linked to their evaluation methodology. The test battery from the BSI namely reuses the four tests of FIPS 140-2 [**63**], which are themselves based on the tests from [**59**] (§5.4.4). In addition to the tests in the SP 800-90 standard (notably [**6**] §5), NIST also offers a set of statistical tests in the SP 800-22 standard [**70**].

In the majority of cases, the aim of these standard tests is to verify the hypothesis that the bits of the sequence are the successive realizations of a *strict-sense stationary discrete process* ([**65**] §6.5.2), following a Bernoulli distribution $\mathscr{B}(\frac{1}{2})$. As a reminder, a strict-sense stationary discrete process is defined as follows:

DEFINITION 3.8 (Strict-sense stationary discrete process). *Let $\{X_i\}$ be a discrete stochastic process, i.e. an object representing the discrete evolution of a random variable $X$ over time. We say that $\{X_i\}$ is strict-sense stationary if and only if, for all $(i,n,k) \in \mathbb{N}^3$, the random vectors $X(i_1,...,i_n)$ (realizations of $\{X_i\}$ at times $i_1,...,i_n$) and $X(i_1+k,...,i_n+k)$ have the same distribution.*

**3.2.4. SP 800-90B standard tests.** Among the statistical tests commonly used today, we find the tests of the SP 800-90B standard ([**6**] §5), which are designed to confront a data sequence the hypothesis that the samples of the sequence are independent and identically distributed (IID). The set of tests is divided into two groups: permutation tests, which apply a statistical measure to permutations of the tested sequence to check that the values of the statistics are normally distributed around the value obtained on the initial sequence, and complementary $\chi^2$ tests.

3.2.4.1. *Permutation tests.* For the test of the first group, the permutation algorithm used is the Fisher-Yates Shuffle [**30**] (Example 12), described in Alg. 3.

**Algorithm 3** Fisher-Yates Shuffle algorithm [**30**]

**Input:** $L > 0, S = (s_1,...,s_L)$.
**Output:** Random permutation of $S$.
  **for** $i = L$ downto 1 **do**
    Generate a random integer $1 \leqslant j \leqslant i$.
    Swap $s_j$ and $s_i$.
  **end for**

*Remark:* The Fisher-Yates algorithm require the use of a first RNG to produce the random index $j$ for the swap between $s_i$ and $s_j$. This RNG is expected to be of good quality, which raises the question of how it was tested in the first place. Its own test procedure would indeed require the use of another RNG, and we fall into an endless loop. It could also be interesting (although not discussed in this manuscript) to evaluate the impact of the use of a poor quality RNG in the Fischer-Yates algorithm on the rest of the test procedure.

Then, in the case of permutation tests, the procedure is the same for all tests:

(1) $C_0, C_1 = 0$.

(2) Compute the test statistic $T$ on the base sequence.

(3) For $i \in \{1, ..., 10{,}000\}$ :
   (a) Build a permutation of the sequence using the Fisher-Yates Shuffle algorithm.
   (b) Compute the test statistic $T'$ on the shuffled sequence.
   (c) If $T' > T$, increment $C_0$, if $T' = T$ increment $C_1$.

(4) If $C_0 + C_1 \leqslant 5$ or $C_0 \geqslant 9995$, reject the hypothesis of an IID sequence.

One permutation test that can be applied to both binary and integer data is the test of number of runs based on the median:

DEFINITION 3.9 (Test of number of runs based on the median). *For a data sequence $\mathscr{S} = (s_i)_{1 \leqslant i \leqslant L}$, the statistic of the number of runs based on the median is computed as such:*

(1) *Compute the median $\overline{X}$ of the dataset.*

(2) *Build a second sequence $\mathscr{S}'$ such that, for any $i \in \{1, ..., L\}$ :*

$$s'_i = \begin{cases} -1 & \text{if } s_i < \overline{X}, \\ +1 & \text{if } s_i \geqslant \overline{X}. \end{cases}$$

(3) *The test statistic on $\mathscr{S}$ is the number of runs in $\mathscr{S}'$, that is to say, the number of subsequences of identical values in $\mathscr{S}'$.*

As an example, for $\mathscr{S} = (5, 27, 31, 4, 1, 12, 2, 68, 54, 13, 22)$, the median $\overline{X}$ is equal to 13, and the sequence $\mathscr{S}'$ is $(-1, +1, +1, -1, -1, -1, -1, +1, +1, +1, +1)$. There are 4 runs in $\mathscr{S}'$ : $(-1)$, $(+1, +1)$, $(-1, -1, -1, -1)$, $(+1, +1, +1, +1)$. The statistic of the number of runs based on the median applied to $\mathscr{S}$ is then equal to 4.

For binary data, the runs can be computed directly on the sequence itself. The runs of numbers based on the median simply become runs of bits 0 and runs of bits 1.

3.2.4.2. $\chi^2$ *tests.* In the case of $\chi^2$ tests, one of the tests applicable to binary data is the test of goodness-of-fit. Its goal is to verify that the distribution of bits 0 and 1 remains the same throughout the sequence.

DEFINITION 3.10 (Test of Goodness-of-fit for binary data). *For a binary sequence* $\mathscr{S} = (s_i)_{1 \leqslant i \leqslant L}$, *the statistic of the test of Goodness-of-fit is computed as follows:*

(1) *Let* $T = 0$ *and* $p = \#\{s_i = 1, i \in \{1, ..., L\}\}$.

(2) *Divide the sequence* $\mathscr{S}$ *into 10 subsequences* $\mathscr{S}_d$ *of equal length* $\left\lfloor \frac{L}{10} \right\rfloor$.

(3) *Define the number of bits* 0 *and* 1 *as being equal to* $e_0 = (1-p)\left\lfloor \frac{L}{10} \right\rfloor$ *and* $e_1 = p\left\lfloor \frac{L}{10} \right\rfloor$.

(4) *Compute the statistic:*

$$T = \sum_{d=1}^{10} \frac{(o_0^{(d)} - e_0)^2}{e_0} + \frac{(o_1^{(d)} - e_1)^2}{e_1},$$

*where* $o_0^{(d)}$ *and* $o_1^{(d)}$ *are the number of bits* 0 *and* 1 *respectively in the subsequence* $\mathscr{S}_d$.

*The test fails, and then the sequence is considered "non-homogeneous" if the value of the statistic is greater than* 27.887, *which is the critical value at* 0.001 *for a* $\chi^2$ *distribution with* 9 *degrees of freedom.*

**3.2.5. AIS 20/31 standard tests.** There are nine AIS 20/31 tests [**43**], all of which (with the exception of test T8) are hypothesis tests, with the null hypothesis being that the tested sequence has been produced by an ideal random number generator. The aim of these tests is to check that the distribution of generated numbers conforms to the hypothesis of an ideally random sequence, particularly with regard to the distribution of $n$-bit words (for a fixed integer $n > 0$) and data independence. More specifically, the tests are the following:

- **Disjointness test (T0) :** Verifies that no 48-bit string is repeated in a sequence of $3{,}145{,}728 = 2^{16} \times 48$ bits.
- **Monobit test (T1) :** Measures the quantity of bits 1 to ensure that there are approximately 50% of them in the sequence.
- **Poker test (T2) :** $\chi^2$ test which aims at ensuring that 4-bit words are evenly distributed.
- **Runs test (T3) :** Aims at ensuring that the number of runs (subsequence of maximum length of identical values) of lengths 1, 2, 3, 4, 5 and $\geqslant 6$ conforms to the hypothesis of an ideally random sequence.
- **Long run test (T4) :** For 20,000 bits, a *long run* is defined as a run of length $\geqslant 34$. The test ensures that no long run is present among the 20,000 bits of the sequence under

study.

- **Autocorrelation test (T5) :** The test studies the correlations between successive bits by verifying that there are not too many pairs of identical or complementary bits.

- **Uniform distribution test (T6) :** Verifies that the distribution of $k$-bit words is uniform in an $n$-bit sequence. Parameters $k, n \in \mathbb{N}^2$ and $a$ (threshold of rejection for the test) can be adjusted. It is not a $\chi^2$, unlike test T2.

- **Test for homogeneity (T7) :** Aims at ensuring that the distribution of $n$-bit words (for a given $n > 0$) stays the same through several successive independent generations of numbers.

- **Entropy estimation (T8) :** Entropy measurement algorithm developed by Coron and Naccache [**20**].

The two tests we will be heavily relying on throughout this manuscript are the Monobit (T1) and Autocorrelation (T5) tests, both already present in [**59**]. These tests are detailed below.

DEFINITION 3.11 (Monobit test). *For a binary sequence $(b_i)$ of length $N \in \mathbb{N}$, the monobit statistic $M$ is defined as follows:*

$$M = \sum_{i=1}^{N} b_i.$$

*In the AIS 20/31, the statistic is applied on a sequence of* 20,000 *bits, and the null hypothesis is that the sequence studied comes from an ideal generator. The test is passed if $M \in [9654, 10346]$.*

In the case of a sequence produced from an ideally random generator, including the assumption of independent data, the expected value and the variance of this statistic are:

$$\mathbb{E}[M] = \frac{N}{2}, \text{ and } \mathrm{Var}(M) = \frac{N}{4}.$$

The Monobit test then allows for deviation of 3.46% from the expected value.

DEFINITION 3.12 (Autocorrelation test). *For a binary sequence $(b_i)$ of length $N \in \mathbb{N}$ and an integer $k \leqslant \frac{N}{2}$, the autocorrelation statistic of lag $k$, denoted by $\mathscr{A}_k$, is computed along the following equation:*

$$\mathscr{A}_k = \sum_{i=k}^{N} b_{i-k} \oplus b_i.$$

*In the AIS 20/31, for a given lag $k$, this statistic is to be applied to sequences of* $5000 + k$ *bits, with the restriction of having $k \leqslant 5000$. The null hypothesis is again that of an ideally random generator, and the test is considered passed if $\mathscr{A}_k \in [2326, 2674]$.*

In the case of a sequence generated by an ideally random generator, which is again the null hypothesis of the test, the expected value and variance of the statistic are:

$$\mathbb{E}[\mathscr{A}_k] = \frac{N-k}{2}, \text{ and } \mathrm{Var}(\mathscr{A}_k) = \frac{N-k}{4}.$$

The test then allows for a deviation of 6.96% from the ideal value.

> *Remark :* In its original iteration in [**59**], the autocorrelation test also uses $\mathscr{A}_k$, but the test statistic, then denoted by $X_5$, is defined as such:
>
> $$X_5 = 2 \times \left( \frac{\mathscr{A}_k - \frac{N-k}{2}}{\sqrt{N-k}} \right).$$
>
> The idea is to normalize the statistic $\mathscr{A}_k$ to get a value that will be comparable to other results, no matter the length $N$ of the tested sequence. It is interesting to note that the statistic $X_5$ is the one that has been chosen as Autocorrelation statistic in the latest draft of the AIS 20/31 [**67**] (§4.6.1, test T4).

3.2.5.1. *Procedure A.* The procedure is similar to the permutation tests in the NIST SP 800-90B standard, in that the tests in the procedure apply both to a base sequence and to modified versions of this base sequence. More precisely, procedure A on a generator producing $n$-bit numbers runs as follows:

(1) Generate a first sequence of 3,145,728 = $2^{16} \times 48$ bits, apply test T0 on it.

(2) Generate a sequence $S_0$ of 20,000 bits, then generate a sequence of 20,000 × $n$ bits. From this second sequence, build $n$ subsequences $(S_i)_{1 \leqslant i \leqslant n}$ of 20,000 bits such that, for any $1 \leqslant i \leqslant n$, $S_i$ is the sequence that contains the $i$-th bit of each of the 20,000 integers. Repeat step (2) until 257 sequences of 20,000 bits have been built.

(3) Apply tests T1 to T4 on each of the 257 sequences built in step (2).

(4) For each of the 257 sequences, apply test T5 as follows: compute the autocorrelation statistic on the first 10,000 bits of the sequences, with lag values $k$ varying between 1 and 5000. Identify the lag $k_{max}$ for which the autocorrelation deviates the most from the expected value of 2500, and compute the autocorrelation statistic with this same lag parameter $k_{max}$, this time on the last 10,000 of the sequence.

3.2.5.2. *Procedure B.* In addition to procedure A, a second procedure, named procedure B is applied as follows on a binary sequence of arbitrary length $N \geqslant$ 2,468,480:

(1) Apply test T6 on the first 100,000 bits of the sequence, with parameters $(k, n, a) = (1, 100{,}000, 0.025)$.

(2) For all remaining bits in the sequence:
   (a) From the pairs of disjoint bits, build two subsequences $TF_0$ and $TF_1$ which respectively contain the pairs for which the first bit is 0 and 1, until both subsequences contain $n_1 \geqslant 100{,}000$ pairs. Compute
   $U_0(1) = \#\{j \leqslant n_1 | (b_{2j+1}, b_{2j+2}) \in TF_0, b_{2j+2} = 1\}/n_1$, the number of pairs for which the first bit is 0 and the second bit is 1, divided by the total number of pairs for which the first bit is 0, namely $n_1$. Similarly, compute $U_1(0)$ the number of pairs $(1,0)$ divided by $n_1$. Finally, compute the value of the test statistic
   $S = |U_0(1) + U_1(0) - 1|$, and verify that $S < 0.02$.

   (b) In a similar fashion, with the next bits in the input sequence, build 4 subsequences of disjoint triplets $TF_{rs}$, $(r,s) \in \{0,1\}^2$ such that $TF_{rs}$ contains the triplets for which the first two bits are $r$ and $s$, until every subsequence contains $n_2 \geqslant 100{,}000$ triplets. Compute the values $U_{rs}(0)$ and $U_{rs}(1)$, the number of triplets $(r,s,0)$ and $(r,s,1)$ divided by $n_2$. Compare the value of $U_{0s}$ to the value of $U_{1s}$ using the $\chi^2$ statistic defined in test T7, with a critical value $\alpha = 0.0001$. We will not detail the test T7, but as an example, for $s = 0$, and after simplifying the expression, computing the statistic of test T7 to compare $U_{0s}$ with $U_{1s}$ amounts to computing the following value:

$$\frac{\big(f(000) - f(100)\big)^2}{f(000) + f(100)} + \frac{\big(f(001) - f(101)\big)^2}{f(001) + f(101)}$$

   where $f(rsx)$ is the frequency of occurrence of the 3-bit word $rsx$, $x \in \{0,1\}$, relatively to the frequency of occurrence of the 2-bit word $rs$.

   (c) Finally, with the next bits in the input sequence, build 8 subsequences of disjoint quadruplets $TF_{rst}$, $(r,s,t) \in \{0,1\}^3$, such that every quadruplet of $TF_{rst}$ starts with bits $r$, $s$ and $t$ in that order, and so that the 8 subsequences are of length $n_3 \geqslant 100{,}000$. For every triplet $(r,s,t) \in \{0,1\}^3$ compute $U_{rst}(0)$ and $U_{rst}(1)$ the frequencies of occurrence of $(r,s,t,0)$ and $(r,s,t,1)$ relatively to $n_3$. Like in the previous step, compare the values of $U_{0st}$ and $U_{1st}$ using the $\chi^2$ statistic defined in test T7, with a critical value $\alpha = 0.0001$.

(3) Apply the test T8 (entropy test) to the rest of the bits in the sequence with parameters $L = 8$, $Q = 2560$ and $K = 256{,}000$. The test is a success if the value of the statistic gets over 7.976, which corresponds to an entropy per bit of 0.997.

In the case of both procedures, the method used to decide whether the procedure has succeeded or failed is the same: if none of the tests fail, the procedure is a success. If two or more tests have failed, the procedure is a failure. If exactly one test fails, the procedure is rerun once

on a completely new dataset (produced, of course, by the same random number generator). If one of the tests fails again, the procedure is deemed a failure. If all tests pass on the second attempt, the procedure is a success.

> *Remark:* While the choice of specific data sizes for the two procedures is somewhat arbitrary, the reasoning behind those choices is that the size of the data must be sufficiently big so that the relative variances of the statistics are low, but not so big that tests become impractical. More specifically, the authors chose the data sizes such that, for an ideal generator, procedure A passes "with probability $\approx 0.9987$", and procedure B passes "with probability $\approx 0.9998$" ([**43**], (210) and (213) p.55-56). In this regard, the test procedures were designed with the primary goal of not rejecting good generators.

**3.2.6. Alternative approach to representing test statistics.** In [**51**] Lubicz, takes up Yao's definition[6] [**80**] of a statistical test on binary sequences as being a probabilistic algorithm taking as an input an $n$-bit sequence, and which returns a binary string which length is lower than $n$.

In this regard, it appears that this definition only describes the statistics used for testing despite the author naming it statistical tests, as there is no mention about regions of acceptance or rejection, which is mandatory for classical statistical tests (see for instance [**47**]). In the remainder of this section, we will then present Lubicz's results as being about statistics rather than tests.

With this definition, Lubicz proves that any finite statistic $F$, applied to binary sequences, can be represented by a finite automaton [**38**] in the form $F = (S, f, s_0)$, where $S = \{s_0, ..., s_k\}$ is the finite set of states of the automaton, $s_0$ is the initial state of the automaton, and $f : S \times \{0,1\} \to S$ is the transition function. To clarify the notion of transition function, $f(s_i, b) = s_j$ means that when, during the test run, we are in state $s_i$ and we read bit $b$, we pass into state $s_j$.

Each statistic therefore works by reading the studied sequence bit by bit, and navigating through the states of the associated automaton using its transition function.

As an example, Lubicz explains that the Monobit statistic (Cf. Def. 3.11) can be computed using the automaton $F = (\{s_0, s_1\}, f, s_0)$, shown in Fig. 13, which is simply the automaton representing a binary memoryless information source [**2**]. To compute the value of the Monobit statistic, all we need to do is count the number of passes through the state $s_1$, which corresponds to the number of bits 1 in the tested sequence.

Similarly, for a given $k \in \mathbb{N}, k > 0$, one can compute the autocorrelation statistic $\mathscr{A}_k$ of the AIS 20/31 (see Def. 3.12) with the $2^{k+1}$-state automaton

[6] [**80**] Def. 12: "A polynomial statistical test is a probabilistic algorithm $M$ that takes only inputs of the form $(x_1, x_2, ..., x_{N^k})$, where each $x_i$ is an $N$-bit number, halts in time $O(N^t)$, and outputs a binary string $y$, where $t$ and $k$ are some fixed positive integers."

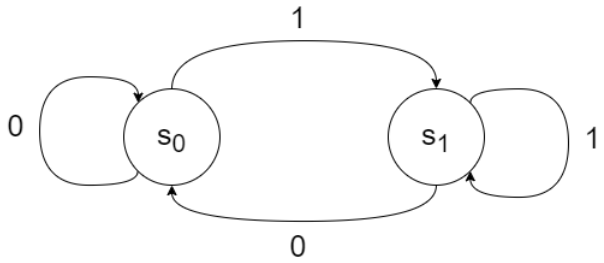



FIGURE 13. Automaton representing a binary memoryless source.

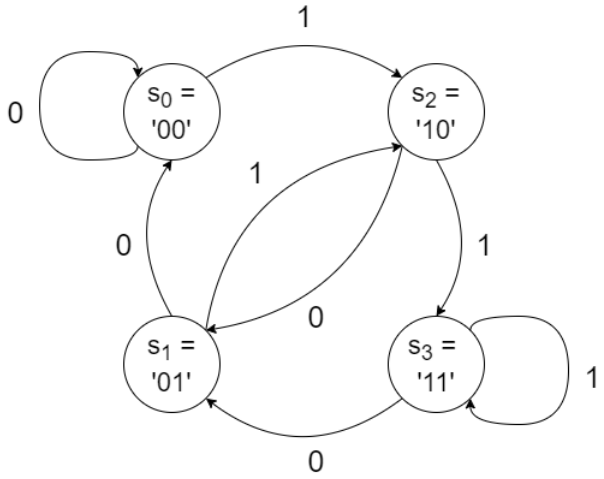



FIGURE 14. Automaton representing a binary source of memory 1.

$F = (\{s_0, ..., s_{2^{k+1}-1}\}, f, s_0)$ in which each state $s_i$ of index $i = \sum_{p=0}^{k} b_p 2^p$ model the reading of the $k+1$ consecutive bits $b_0, ..., b_k$, and such that $f(s_i, b) = s_j$ where $b \in \{0,1\}$ and $j = \sum_{p=1}^{k} b_p 2^{p-1} + b \times 2^k$. In other words from the state which index has as a binary representation the string $b_k \ldots b_1\ b_0$, the transition function leads, upon reading the bit $b$, to the state which index has as a binary representation the string $b\ b_k \ldots b_1$.

This automaton is in fact the automaton modeling a binary source of memory $k$ [**2**].

As an example, for $k = 1$, which corresponds to the study of directly successive bits in the sequence, the automaton is represented in Fig. 14. The statistic $\mathscr{A}_1$ is then computed by summing the number of passes through the states $s_1$ and $s_2$.

In the general case, the statistic $\mathscr{A}_k$ is computed by summing the number of passages in all states of index $i = \sum_{p=0}^{k} b_p 2^p$ such that $b_0 \neq b_k$.

> *Remark:* To be entirely rigorous, the automaton shown in Fig. 14 would have had to be supplemented by two additional states, modeling the reading of the first bit of the sequence. But for the sake of clarity, we have chosen to consider that the automaton is initialized by reading the first two bits of the sequence at the same time to place itself in one of the 4 states depicted in the graph.

This formalism of statistics makes it possible to show, in particular, an intuitive result: the application of a **battery of statistical tests** on a sequence will give a guarantee on the quality of the randomness **at least as good** as applying **only part of this battery**.

Indeed, in terms of automata, the application of several joint statistics corresponds to passing the sequence through the product automaton of the automata corresponding to each individual statistic ([**51**], Def. 29). Lubicz then explains that isolating a statistic (or, equivalently, several statistics) from this product automaton corresponds to applying a morphism (more specifically a projection) to it.

Furthermore, Lubicz defines an order relation ([**51**], Def. 24) on statistics, such that, if there exists a morphism from the statistic $F$ to the statistic $F'$, then $F$ is stronger than $F'$, which he denotes by $F \geqslant F'$. He then shows ([**51**], Cor. 28) that if $F \geqslant F'$, then "*any sequence that passes the test $F$ also passes the test $F'$*".

> *Note:* To be applicable in the sense of statistical tests as we defined them in subsection 3.2.1, the notion of acceptance region needs to be added to the above corollary to be able to compare two tests based on the statistics $F$ and $F'$.

By combining all these results, the author formally demonstrates, as announced, that a test battery will provide at least as good a guarantee as any of its individual tests on the quality of a sequence's randomness.

And although this alternative vision of test statistics does not necessarily lead to the development of specific statistics, it does provide a basis for a formalism which we shall continue to refer to in the remainder of this manuscript as it is useful to compare statistics in particular.

## 3.3. Limitations of statistical tests

Although statistical tests allow for an "affordable" estimation of the quality of a random number generator through the statistical properties of the data it produces, these tests still suffer from a number of drawbacks.

**3.3.1. No strict guarantee on the security of a tested RNG.** By construction, statistical tests only offer a binary answer on whether an RNG has acceptable statistical properties or not, based on a limited sample of data. This result is moreover subject to false positives and false negatives (also called Type I and II errors, see Sect. 3.2.2). In that sense, the result of a test is only to be seen as an indicator of the general quality of a generator, with a certain confidence level, but it **cannot constitute a strict proof of the security** of said generator. More specifically, a generator cannot be deemed usable for a cryptographic application based only on the result of a test battery.

> *Note:* Although from a certification standpoint, statistical tests cannot provide a definite guarantee on the quality of a generator, they can lead to the detection to statistical anomalies in the generated data. These anomalies, can then be exploited to lead to various attacks. As an example, if statistical tests showcase that some of the bits produced by the RNG are strongly correlated, and could thus be deduced from one another, attacks such as the one presented by Nguyen and Shparlinski [**60, 61**] could be attempted.

**3.3.2. Low restrictiveness of tests.** To limit the risk of type I or II errors, test developers will then adapt the range of values of the statistic for which the test is considered passed. For example, in the case of AIS 20/31, the statistical rejection ranges of tests T1 to T5 are set to fix the type II error (false negative) at a value of $10^{-6}$. The aim is to never reject a good-quality generator. However, Type I error (false positive) is not discussed, and it can be argued that the bounds of some tests might be too lax. For the Monobit (T1) test, for example, a sequence producing 51% bits 1 would likely pass the test, which could pose a problem depending on the desired level of security. Of course, with such binary tests, a compromise is necessary, as it is impossible to minimize both Type I and Type II errors.

**3.3.3. Redundancy of tests.** A second limitation of these test batteries is the redundancy of some of them. As we explained in subsection 3.2.6, an abundance of tests will not impair the accuracy of the evaluation.

On the other hand, an accumulation of statistical tests can also complexify test procedures unnecessarily by adding redundancy in the statistical characteristics that are covered by the tests. For example, as mentioned by De Julis ([**21**] section 3.2.4), with the parameters recommended by procedure B of AIS 20/31 [**43**], the T6 test is completely redundant with the Monobit test (T1). More precisely, in the general case, the T6 test is expressed as:

DEFINITION 3.13 (Test T6 form the AIS 20/31). *For $(k,n) \in \mathbb{N}^2$, $a \in \mathbb{R}, a \geqslant 0$, and a sequence of $n$ words of $k$ bits $\mathscr{S} = (w_1, ..., w_n)$, the statistic of test T6 is defined as:*

$$T_6(x) = \frac{\#\{j \in \{1, ..., n\} | w_j = x\}}{n},$$

*and the test is passed of, for all $x \in \{0,1\}^k$, $T_6(x) \in [2^{-k} - a, 2^{-k} + a]$.*

In Procedure B, the test parameters are set to $k = 1$ and $n = 100{,}000$. In practice, the test is therefore used to study the proportion of bits 0 and 1 in the sequence, similarly to the Monobit test (Def. 3.11) apart from the number of bits of the tested sequence, but with a more permissive range. Indeed, the accepted deviation from the ideal value for the Monobit test is 3.46%, while the accepted deviation for the T6 test is 5%, despite the greater quantity of data studied (and therefore the normally lower variance of the statistic). In the new version of AIS 20/31 [**67**] (currently under review), the T6 test has been removed.

**3.3.4. Difficulty in interpreting test results.** Finally, a third limitation of statistical tests as used in current standards is the difficulty of interpreting results in certain cases. More specifically, in a general context, the link between a test result and the statistical characteristic it is intended to study is not straightforward. A striking example of this is the autocorrelation test (T5) from the AIS 20/31. For a stationary process (Cf. Def. 3.8), which is the framework for the AIS 20/31 tests, statistical autocorrelation is a well-defined notion:

DEFINITION 3.14 (Autocorrelation of a stationary discrete process $\{X_i\}$). *For a stationary discrete stochastic process $\{X_i\}$, the autocorrelation of lag $k > 0$ of $\{X_i\}$, denoted by $\rho_k$, is defined by the following equation:*

$$\rho_k = \frac{\mathbb{E}\left[(X_i - \mu)(X_{i-k} - \mu)\right]}{\mathrm{Var}\,(X_i)},$$

*where $\mu = \mathbb{E}[X_i]$ is the expected value of the process $\{X_i\}$.*

A priori, it is reasonable to expect that the autocorrelation test will focus on the estimation of the autocorrelation of the tested sequence. Even more so considering that, in the new draft of the AIS 20/31, the authors specifically mention that the test suite in which the autocorrelation test is used **focuses on the security property PTG.2.2** ([**67**], (279) p.45), which deals with **dependencies only**. However, it quickly becomes apparent that this test is impacted by statistical anomalies other than a simple autocorrelation. Let us take, for example, the case of a sequence $\mathscr{S} = (s_i)$ produced by the following rule: for $N \in \mathbb{N}$, for any $1 \leqslant i \leqslant N$, $\Pr(S_i = 1) = 0.8$, with $S_i$ the random variable representing the bits $s_i$ of the sequence. The sequence is then constructed using $N$ independent random drawings to determine the value of successive bits according to this rule.

For such a sequence, the expected value of the autocorrelation statistic $\mathscr{A}_k$ of test T5 is equal to:

$$\mathbb{E}[\mathscr{A}_k] = \sum_{i=k}^{5000+k} \mathbb{E}[S_{i-k} \oplus S_i] = 5000 \times 2 \times (0.8 \times 0.2) = 1600.$$

for any $k \in \{1, \ldots, 5000\}$. The test is than almost guaranteed to fail.

However, for this sequence, using the notations of Def.3.14 and using the fact that $\Pr(S_i = 1) = \mu$, that the drawings of the values of $S_i$ are independent and that $S_i$ is stationary, we also have the following equation:

$$\begin{aligned}
\mathbb{E}\left[(S_i-\mu)(S_{i-k}-\mu)\right] = \quad & (1-\mu)(0-\mu)\times \Pr(S_i=1, S_{i-k}=0) \\
& +(0-\mu)(1-\mu)\times \Pr(S_i=0, S_{i-k}=1) \\
& +(1-\mu)(1-\mu)\times \Pr(S_i=1, S_{i-k}=1) \\
& +(0-\mu)(0-\mu)\times \Pr(S_i=0, S_{i-k}=0), \\
= \quad & -2\times\mu^2(1-\mu)^2+2\times\mu^2(1-\mu)^2, \\
= \quad & 0.
\end{aligned}$$

Then, the **theoretical autocorrelation** of that sequence is **null**, but the autocorrelation test from the AIS 20/31 still **fails**.

In this sense, the result of this test may be misleading as to the statistical feature of the sequence that is leading to the failure, which is, in that case, a general disproportion between bits 0 and 1, and not an autocorrelation.

> *Remark:* The autocorrelation statistic $X_5$ used in the new draft of the AIS 20/31 also suffers from the same drawback, as it is simply a normalized version of the statistic $\mathscr{A}_k$ (see [**67**], §4.6.1, test T4).

Although not as striking as in the case of test T6, it also appears that the autocorrelation test is partially redundant with the Monobit test, considering that it is affected by a statistical anomaly that is perfectly characterized by the Monobit statistic.

This overlap in analyzed characteristics between tests could be the result of an overly superficial analysis of the anomalies that the tests aim at studying. More precisely, the tests are established empirically, with a complete modeling of their behavior in the event that they are applied to an ideally random sequence of numbers, but without a prior characterization of the anomalies the tests aim at covering.

In the context of an evaluation, aimed at verifying that the generator meets the announced security level, the lack of a direct link between a test failure and the statistical anomaly at cause might not be a problem, as long as a poor quality generator has very little chance of passing the tests. Although evaluators often look to verify a single property of the generator, linked to its stochastic model. For example, the evaluators might know from the model that the generator suffers from a frequency anomaly, and want to verify whether it also presents a correlation anomaly. In this event, a failure of the autocorrelation test from the AIS 20/31 might mislead the evaluator into thinking that correlations are also present in the generated data.

From the point of view of a designer of a random number generator, having a means to trace back to the statistical anomaly that led to a given failure is obviously very interesting, as it could enable to identify the point(s) to be corrected in the generator's design, which may vary according to the type of defect encountered. Statistical anomaly characterization tools could

even enable a TRNG designer to verify the law followed by his generator, even in the case that this law is not the typical uniform distribution which is expected for random number generators.

As an example, in the case of TRNGs based on resistive memories (called RRAM) (see subsection 2.5.3), such as OxRAM [**66, 3**], a global disproportion of bits 0 and 1 may be the result of a poor choice of decision threshold for the resistance of cells distinguishing a high resistivity state from a low resistivity state. A distinction threshold that is too low could lead to a preponderance of bits 1, for example. Autocorrelation in the generated data could, in turn, be the result of a defective reset of the memory cells (again, see 2.5.3 for more details of how such generators work).

In this sense, as well as enabling the rejection of poor-quality generators, it seems interesting for a battery of tests to focus on the study of precisely defined statistical characteristics, avoiding redundancies if possible in order to make the interpretation of their results more straightforward.

## 3.4. Modeling statistical anomalies

The statistical features we are interested in when evaluating the quality of a random number generator are those that distinguish it from an ideally random generator. In the remainder of this manuscript, we will refer to these distinguishing features as **statistical anomalies**, since they represent statistical phenomena that are not found in the case of an ideally random generator. These statistical anomalies are referred to as "defect in the randomness" in the PG-083 guide ([**1**] 2.4.2). A precise definition of some anomalies could then further guide a designer to choose the relevant tests for the evaluation of his random number generator, depending on which anomalies they expect their generator to be subject to.

In order to study anomalies in detail, a model in necessary to allow their characterization. More specifically, it seems interesting to construct anomalies as perturbations of the ideal model, to which one or more parameters have been added to account for the anomalies.

To characterize the statistical anomaly, it is necessary to design an estimator for the significant parameters of the model, which can then be used as the basis for creating a test specifically adapted to this anomaly. In this sense, we define the notion of optimality of a statistical test with respect to a given anomaly:

DEFINITION 3.15 (Optimal statistical test for a given anomaly). *For a given statistical anomaly, characterized by a parameter $\alpha$, let $X$ be a statistic and $T_{X,S_0}$ a statistical test associated to $X$. We say that $T_{X,S_0}$ is **optimal for studying the anomaly characterized by** the parameter $\alpha$ if and only if $X$ is an unbiased estimator of $\alpha$, in other words, if:*

$$\mathbb{E}[X] = \alpha.$$

In the remainder of this manuscript, we will then be applying this methodology in an attempt to accurately characterize the two statistical anomalies that we already mentioned: the non-uniformity of the 0 and 1 bit generation frequency, and the correlation between the bits of the studied sequence. In particular, the model for the correlation anomaly should enable us to solve the problem of overlap between the Monobit and Autocorrelation tests of the AIS 20/31, with the introduction of a new autocorrelation statistic. We will prove that this statistic allows for the definition of an optimal test, and will be designed to not be impacted by a global disproportion of bits 0 and 1 in particular.

CHAPTER 4

# Frequency anomaly

*Preliminary remark :* In the rest of the manuscript, unless otherwise stated, the processes considered will all be strict-sense stationary (see [**65**] §6.5.2, or our Def. 3.8).

Also, for the sake of conciseness of expressions, we will use the following notations to describe the probabilities of observing specific bits in the sequence under study:

- $\Pr(B_i = x) = \Pr_i(x)$
- $\Pr\left(B_i = x, B_j = y\right) = \Pr_{i,j}\left(x, y\right)$
- $\Pr\left(B_i = x \mid B_j = y\right) = \Pr_{i,j}\left(x \mid y\right)$

where $\{B_i\}$ is the stochastic process modeling the bits of the sequence.

The first anomaly we are looking to characterize in binary sequences produced by random number generators is the disproportion between the frequency of occurrence of bits 0 and 1, which we will call the **frequency anomaly**. The aim is to model the anomaly by a function of a parameter $\xi$, which is symmetrical and normed (with values in $]-1,1[$) such that the case $\xi = 0$ represents the case where the sequence presents no anomaly.

Under these conditions, the model we have chosen is the following:

DEFINITION 4.1 (Frequency model for the bit 1). *For a binary sequence $(b_i)_{1 \leqslant i \leqslant N}, N \in \mathbb{N}$, seen as the successive realizations of a binary stochastic process $\{B_i\}$, we model the frequency of occurrence of the bit* 1 *as such:*

$$\Pr_i(1) = \frac{1+\xi}{2},$$

*where* $\xi \in ]-1,1[$.

By construction, this model also enables the characterization of the frequency of occurrence of the bit 0, as we naturally have $\Pr_i(0) = \frac{1-\xi}{2}$.

It would of course be mathematically correct to consider that $\xi \in [-1,1]$. However, we will later see that it is easier to exclude the two extreme cases of having $\xi \in \{-1,1\}$, and simply

consider them as limit cases of our model if needed.

To characterize the frequency anomaly, we must then be able to estimate the parameter $\xi$ linked to the model. To this end, the most efficient estimator turns out to be the AIS 20/31 Monobit statistic (Def. 3.11). Indeed, for a sequence $(b_i)_{1\leqslant i\leqslant N}$ of realizations of the binary process $\{B_i\}$, the expected value Monobit statistic $M$ is:

$$\mathbb{E}[M] = \mathbb{E}\left[\sum_{i=1}^{N} B_i\right] = \sum_{i=1}^{N} \mathbb{E}[B_i] = N \times \Pr_i(1) = N \times \frac{1+\xi}{2}.$$

Thus, an interesting statistic for the evaluation of the frequency anomaly can be defined as follows:

DEFINITION 4.2 (Statistic for the frequency anomaly). *Let $M$ be the Monobit statistic of the AIS 20/31 and $N \in \mathbb{N}$ be the number of bits in the studied sequence. We define the statistic $\hat{\xi}$ as:*

$$\hat{\xi} = \frac{2}{N} \times M - 1.$$

THEOREM 4.3. *The statistic $\hat{\xi}$, established in Def. 4.2, is an unbiased estimator of $\xi$.*

PROOF. Using the expression of the expected value of the Monobit statistic $M$, we directly have:

$$\mathbb{E}\left[\hat{\xi}\right] = \frac{2}{N} \times \mathbb{E}[M] - 1 = \xi.$$

$\hat{\xi}$ is therefore indeed an unbiased estimator of $\xi$.

□

In this sense, the statistic $\hat{\xi}$ enables us to define an optimal test (Cf. Def. 3.15) for the frequency anomaly as modeled above, and the Monobit test proposed by the AIS 20/31 is therefore almost optimal, needing just a linear transformation.

Furthermore, if we denote $B_i$ the random variable modeling the observation of the process $\{B_i\}$ at a given time $i$, the variance of the statistic $\hat{\xi}$ is:

$$\begin{aligned}\mathrm{Var}\left(\hat{\xi}\right) &= \frac{4}{N^2} \times \mathrm{Var}\left(M\right), \\ &= \frac{4}{N^2} \times \mathrm{Var}\left(\sum_{i=1}^{N} B_i\right), \\ &= \frac{4}{N^2} \times \left(\sum_{i=1}^{N} \mathrm{Var}\left(B_i\right) + 2 \times \sum_{j=2}^{N} \sum_{k=1}^{j-1} \mathrm{Cov}(B_j, B_{j-k})\right).\end{aligned}$$

And, according to our model, $B_i$ follows a Bernoulli distribution with parameter $\frac{1+\xi}{2}$, so:

$$\mathrm{Var}\left(B_i\right) = \frac{1-\xi^2}{4}$$

However, our model for the frequency anomaly does not enable us to characterize the covariance of the pair $(B_j, B_{j-k})$, and it appears that the variance of the statistic $\hat{\xi}$ is affected by the presence of a potential autocorrelation in the sequence. Using the term $\rho_k$ introduced in Def. 3.14, we can establish the following expression for the variance of the statistic $\hat{\xi}$:

PROPERTY 4.4 (Variance of the statistic $\hat{\xi}$). *Let $N \in \mathbb{N}$ be the number of bits in the tested sequence and $\rho_k$ be the autocorrelation of lag $k$ in the sequence. Then the variance of the $\hat{\xi}$ statistic is expressed as follows:*

$$\mathrm{Var}\left(\hat{\xi}\right) = (1-\xi^2) \times \left(\frac{1}{N} + \frac{2}{N^2} \times \sum_{j=2}^{N} \sum_{k=1}^{j-1} \rho_k\right).$$

PROOF. We have already established that:

$$\begin{aligned}\mathrm{Var}\left(\hat{\xi}\right) &= \frac{4}{N^2} \times \left(\sum_{i=1}^{N} \mathrm{Var}\left(B_i\right) + 2 \times \sum_{j=2}^{N} \sum_{k=1}^{j-1} \mathrm{Cov}(B_j, B_{j-k})\right), \\ &= \frac{1-\xi^2}{N} + \frac{8}{N^2} \times \sum_{j=2}^{N} \sum_{k=1}^{j-1} \mathrm{Cov}(B_j, B_{j-k}).\end{aligned}$$

And, by definition of the covariance:

$$\mathrm{Cov}(B_j, B_{j-k}) = \mathbb{E}\left[(B_j - \mathbb{E}\left[B_j\right])(B_{j-k} - \mathbb{E}\left[B_{j-k}\right])\right].$$

Here again, the process $\{B_j\}$ is stationary. We thus have $\mathbb{E}\left[B_j\right] = \mathbb{E}\left[B_{j-k}\right]$, which we simply write $\mu$. As a reminder the autocorrelation of lag $k$ of $B_j$ is equal to:

$$\rho_k = \frac{\mathbb{E}\left[(B_i - \mu)(B_{i-k} - \mu)\right]}{\mathrm{Var}\left(B_i\right)},$$

which means that:

$$\begin{aligned}\mathrm{Cov}(B_j, B_{j-k}) &= \mathrm{Var}\,(B_i) \times \rho_k, \\ &= \frac{1-\xi^2}{4} \times \rho_k.\end{aligned}$$

Thus, as announced, we have:

$$\mathrm{Var}\left(\hat{\xi}\right) = (1-\xi^2) \times \left(\frac{1}{N} + \frac{2}{N^2} \times \sum_{j=2}^{N} \sum_{k=1}^{j-1} \rho_k\right).$$

□

CHAPTER 5

# Correlation anomaly

## Contents



With the frequency anomaly perfectly characterized, we now bring our focus to a second interesting statistical anomaly: the presence of correlation between the bits of a sequence.

For a proper analysis of this anomaly, we must find the model that best characterizes it. The correlation is an image of the information one can gather on the future bits based on the knowledge of previously generated data. For our model, it thus seems intuitive to make use of the conditional probability $\Pr_{i,i-k}\left(b \mid \overline{b}\right)$, where $k \in \mathbb{N}, k > 0$ and $b \in \{0,1\}$ ($\overline{b}$ being the complementary of the bit $b$).

Before defining a precise model based on this probability, it is interesting to note that these models will all be linked to the model of the frequency anomaly through the following property:

PROPERTY 5.1 (Link between the frequency and correlation anomalies). *Let $\{B_i\}$ be the stationary stochastic process describing the bits of the sequence under study, and $k \in \mathbb{N}, k > 0$. We have the following equation:*

$$\Pr_i(1) \times \Pr_{i,i-k}(0 \mid 1) = \Pr_i(0) \times \Pr_{i,i-k}(1 \mid 0).$$

PROOF. Let $k \in \mathbb{N}, k > 0$. We have, in all generality:

$$\Pr_i(1) = \Pr_{i,i-k}(1 \mid 1) \times \Pr_{i-k}(1) + \Pr_{i,i-k}(1 \mid 0) \times \Pr_{i-k}(0).$$

The processus $\{B_i\}$ being stationary, $\Pr_{i-k}(1) = \Pr_i(1)$ and $\Pr_{i-k}(0) = \Pr_i(0)$. This leads to:

$$\Pr_i(1) \times \left(1 - \Pr_{i,i-k}(1 \mid 1)\right) = \Pr_{i,i-k}(1 \mid 0) \times \Pr_i(0),$$

Or, as announced:

$$\Pr_i(1) \times \Pr_{i,i-k}(0 \mid 1) = \Pr_i(0) \times \Pr_{i,i-k}(1 \mid 0).$$

□

## 5.1. Modeling the correlations between the bits of a sequence

Very similarly to the model we developed to characterize the frequency anomaly, our objective here is to propose a model for the correlation anomaly between bits distant of $k > 0$ (so between $b_1$ and $b_{k+1}$, $b_2$ and $b_{k+2}$, and so on) as the most simple function of a parameter $\delta_k$ which lives in $]-1,1[$. Again, we want that the case where $\delta_k = 0$ depicts the fact that the sequence presents no correlation, and that $\delta_k \to 1$ and $\delta_k \to -1$ depict the situation where the bits distant of $k$ all have the same value, or complementary values respectively.

However, as we will show later, a model of the correlations using conditional probabilities $\Pr_{i,i-k}\left(b \mid \overline{b}\right)$ must take into account the frequency anomaly, and therefore be function of the parameter $\xi$, in addition to our new parameter $\delta_k$.

As with the model for the frequency anomaly, we wanted our model of the correlation anomaly to be polynomial. Taking all these constraints into account, we show that the simplest model for describing the correlations between bits distant of $k > 0$ in a sequence is the following:

DEFINITION 5.2 (Bivariate model for the correlations between bits of a sequence). *For $i \in \mathbb{N}$, $k > 0$, we model the correlation between the bit of index $i-k$ and the bit of index $i$ by the following set of equations:*

$$\Pr_{i,i-k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_k)}{2},$$

*and*

$$\Pr_{i,i-k}(1 \mid 0) = \frac{(1+\xi)(1-\delta_k)}{2},$$

*where* $\xi \in ]-1,1[$ *is the parameter of the model for the frequency anomaly defined in Def. 4.1, and* $\delta_k \in ]-1,1[$ *is the parameter which describes the amplitude of the correlation anomaly between bits distant of* $k > 0$.

To show that this model is optimal for the characterization of the correlation anomalies, we will first demonstrate that the model must indeed take into account the parameter $\xi$ characteristic of the frequency anomaly. Next, we will demonstrate why, in addition to needing to be bivariate, our polynomial model must be of degree 2. Finally, we will show that, the unique polynomial of degree 2 which satisfies our set of constraints is the one we propose in Def. 5.2.

**5.1.1. The necessity to have a bivariate polynomial.** First of all, in all generality, the characteristic term for the transition from a bit 1 to a bit 0 does not have to be equal to the characteristic term for the transition from a bit 0 to a bit 1. This means that, in general, in our model, the conditional probabilities $\Pr_{i,i-k}(0 \mid 1)$ and $\Pr_{i,i-k}(1 \mid 0)$ should be functions of some parameters $\delta_{k,0}$ and $\delta_{k,1}$ respectively.

Secondly, it can be established that a model of the conditional probabilities with only $\delta_{k,0}$ and $\delta_{k,1}$ as parameters cannot properly characterize a correlation anomaly. Indeed, we want the case $\delta_{k,0} = 0$ or $\delta_{k,1} = 0$ to represent the absence of a correlation anomaly between bits distant of $k > 0$. So, in the hypothesis of a polynomial model in the only parameter $\delta_{k,0}$ for $\Pr_{i,i-k}(0 \mid 1)$ for example, written as a function $f_0 : \delta_{k,0} \mapsto f_0(\delta_{k,0})$, we have:

$$f_0(0) = \Pr_i(0) = \frac{1-\xi}{2}.$$

This equation must be verified for any value of $\xi$, so $f_0(0)$ cannot be a constant, and the hypothesis of a monovariate model in $\delta_{k,0}$ cannot be valid. The model must therefore be at least bivariate.

**5.1.2. The necessity to have a polynomial model of degree 2.** In addition to the necessity to include the characteristic term of the model for the frequency anomaly into the model for the correlation anomaly, we can also demonstrate the polynomial of our model must be of degree at least 2. Indeed, if we make the hypothesis of a polynomial of degree 1 in $\xi$ and $\delta_k$, we have, in all generality:

$$\Pr_{i,i-k}(0 \mid 1) = a_0 + b_0\xi + c_0\delta_{k,0},$$

and

$$\Pr_{i,i-k}(1 \mid 0) = a_1 + b_1\xi + c_1\delta_{k,1}.$$

As a reminder, the cases $\delta_{k,0} = 0$ and $\delta_{k,1} = 0$ correspond to cases where only the frequency

anomaly affects the sequence. We then necessarily have $a_0 = a_1 = b_1 = \frac{1}{2}$ and $b_0 = -\frac{1}{2}$.

Also, we recall that Prop. 5.1 links the two conditional probabilities together, and provides here:

$$\frac{1+\xi}{2} \times \left( \frac{1-\xi}{2} + c_0 \delta_{k,0} \right) = \frac{1-\xi}{2} \times \left( \frac{1+\xi}{2} + c_1 \delta_{k,1} \right),$$

which can be simplified into:

$$\frac{1+\xi}{2} \times c_0 \delta_{k,0} = \frac{1-\xi}{2} \times c_1 \delta_{k,1},$$

because $\xi \notin \{-1, 1\}$.

It then appears that the terms $\delta_{k,0}$ and $\delta_{k,1}$ are bound to one another, and the initial set of equation can be rewritten as such:

$$\Pr_{i,i-k}(0 \mid 1) = \frac{1-\xi}{2} + c_0 \delta_{k,0},$$

and

$$\Pr_{i,i-k}(1 \mid 0) = \frac{1+\xi}{2} + \frac{1+\xi}{1-\xi} c_0 \delta_{k,0}.$$

Under our set of constraints, the two equations of the model are then strongly tied together and cannot be both polynomial of degree 1 in $\xi$ and $\delta_{k,0}$ or $\delta_{k,1}$. It is therefore necessary for the model for the correlation anomaly to be of degree at least 2.

**5.1.3. Proof of the unicity of our correlation model.** In an attempt, to find an expression of our model in the form of a polynomial of degree 2, we started from its general expression, and identified all the coefficients using the constraints we had set. We specifically want that the model's parameters vary within $]-1, 1[$, that $\xi = 0$ corresponds to the case where the sequence is subject to no frequency anomaly, and $\delta_{k,0} = 0$ and $\delta_{k,1} = 0$ correspond to the cases where the sequence is affected by no correlation anomaly.

For the model of $\Pr_{i,i-k}(0 \mid 1)$ for example, the general expression is the following:

$$\Pr_{i,i-k}(0 \mid 1) = a_1 \delta_{k,0}^2 + a_2 \xi^2 + a_3 \delta_{k,0} \xi + b_1 \delta_{k,0} + b_2 \xi + c,$$

where $(a_1, a_2, a_3, b_1, b_2, c) \in \mathbb{R}^6$.

When $\delta_{k,0} = 0$, so in the absence of correlation, we want to retrieve the model defined in 4.1 for the frequency anomaly, which provides:

$$\Pr_i(0) = \frac{1-\xi}{2} = a_2\xi^2 + b_2\xi + c.$$

Therefore, by identification of the coefficients, $a_2 = 0, b_2 = -\frac{1}{2}$ and $c = \frac{1}{2}$.

When $\xi = 0$, so in the absence of any frequency anomaly, we want to find the simplest possible correlation model, typically a linear model, with the following constraints: $\delta_{k,0} \in ]-1,1[$, $\delta_{k,0} = 0$ depicts the case where no correlation anomaly is present, $\delta_{k,0} \mapsto 1$ the case where bits distant from $k$ systematically take the same value, and $\delta_{k,0} \mapsto -1$ the case where they systematically take complementary values. More precisely, for $\xi = 0$, we then want to have:

$$\Pr_{i,i-k}(0 \mid 1) = \frac{1-\delta_{k,0}}{2} = a_1\delta_{k,0}^2 + b_1\delta_{k,0} + c,$$

which confirms that $c = \frac{1}{2}$ and provides $a_1 = 0$ and $b_1 = -\frac{1}{2}$.

Finally, to identify the last coefficient, we used a reasoning at the limits on $\delta_{k,0}$ (which can be done thanks to polynomials being continuous functions). More specifically, with our constraint that, when $\delta_{k,0} \to 1$, bits distant of $k$ will almost surely (in the probabilistic sense) have the same value, we have:

$$\lim_{\delta_{k,0}\to 1} \Pr_{i,i-k}(0 \mid 1) = 0 = a_3\xi - \frac{1}{2} - \frac{1}{2}\xi + \frac{1}{2},$$

which provides $a_3 = \frac{1}{2}$ and further confirms the other coefficients.

We then applied the same reasoning to obtain the expression for $\Pr_{i,i-k}(1 \mid 0)$ and thus end up with the following model for the correlation anomaly:

$$\Pr_{i,i-k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_{k,0})}{2},$$

and

$$\Pr_{i,i-k}(1 \mid 0) = \frac{(1+\xi)(1-\delta_{k,1})}{2}.$$

Again, Prop. 5.1 forces the parameters $\delta_{k,0}$ and $\delta_{k,1}$ to be tied to one another. More specifically, with the model we propose, Prop. 5.1 translates into:

$$\frac{1+\xi}{2} \times \frac{(1-\xi)(1-\delta_{k,0})}{2} = \frac{1-\xi}{2} \times \frac{(1+\xi)(1-\delta_{k,1})}{2},$$

which means that $\delta_{k,0} = \delta_{k,1}$ as $\xi \notin \{-1,1\}$. In the remainder of this manuscript, we will then

simply write this common term $\delta_k$.

All in all, we have then proven the existence and uniqueness of the model for the correlation anomaly established in Def. 5.2, in the form of two polynomial equations of degree 2, which describe the correlations through a single parameter $\delta_k$, independently of the value of the individual bits encountered.

## 5.2. Link between the correlation model and theoretical autocorrelation

To further validate our choice of model, it is interesting to compare it with the theoretical autocorrelation of a stationary process. As a reminder the expression for this theoretical autocorrelation of lag $k > 0$, denoted by $\rho_k$, already defined in defined in Def. 3.14:

$$\rho_k = \frac{\mathbb{E}\left[(X_i - \mu)(X_{i-k} - \mu)\right]}{\mathrm{Var}\,(X_i)},$$

where $\{X_i\}$ is a discrete stationary stochastic process and $\mu$ is its expected value.

In our case, the stationary discrete process is the process $\{B_i\}$ which describes the bits of the tested sequence. We then have the following theorem:

THEOREM 5.3. *Let $\{B_i\}$ be a stationary discrete process, with values in $\{0,1\}$, which is described by the model established in Def. 5.2. The theoretical autocorrelation of $\{B_i\}$ is then equal to:*

$$\rho_k = \delta_k$$

PROOF. Let $B_i$ be the random variable describing the observation of the stochastic process $\{B_i\}$ at a given time. Since $\{B_i\}$ takes values in $\{0,1\}$, this variable follows a Bernoulli distribution. According to our model for the frequency anomaly established in Def. 4.1, we then have:

$$\mathrm{Var}\,(B_i) = \Pr_i(1) \times \Pr_i(0) = \frac{1-\xi^2}{4},$$

and

$$\mu = \mathbb{E}[B_i] = \Pr_i(1) = \frac{1+\xi}{2}.$$

Furthermore, we also have:

$$\begin{aligned}\mathbb{E}\left[(B_i - \mu)(B_{i-k} - \mu)\right] = {} & (1-\mu)(1-\mu) \times \Pr_{i,i-k}(1 \mid 1) \Pr_{i-k}(1) \\ & + (0-\mu)(1-\mu) \times \Pr_{i,i-k}(0 \mid 1) \Pr_{i-k}(1) \\ & + (1-\mu)(0-\mu) \times \Pr_{i,i-k}(1 \mid 0) \Pr_{i-k}(0) \\ & + (0-\mu)(0-\mu) \times \Pr_{i,i-k}(0 \mid 0) \Pr_{i-k}(0),\end{aligned}$$

or, using the hypothesis of stationarity to replace $\Pr_{i-k}(x)$, $x \in \{0,1\}$ by $\Pr_i(x)$, and replacing $\mu$ by $\Pr_i(1)$:

$$\begin{aligned}\mathbb{E}\big[(B_i-\mu)(B_{i-k}-\mu)\big] = \Pr_i(1)\Pr_i(0) \times \Big[ &\Pr_i(0)\Pr_{i,i-k}(1\mid 1) - \Pr_i(0)\Pr_{i,i-k}(1\mid 0) \\ &+ \Pr_i(1)\Pr_{i,i-k}(0\mid 0) - \Pr_i(1)\Pr_{i,i-k}(0\mid 1)\Big],\end{aligned}$$

By noticing that $\Pr_i(1)\Pr_i(0)$ is exactly the expression of the variance of $B_i$, the expression of $\rho_k$ can then be simplified as such:

$$\begin{aligned}\rho_k = \quad & \Pr_i(0)\big[\Pr_{i,i-k}(1\mid 1) - \Pr_{i,i-k}(1\mid 0)\big] \\ & + \Pr_i(1)\big[\Pr_{i,i-k}(0\mid 0) - \Pr_{i,i-k}(0\mid 1)\big], \\ = \quad & \big(\Pr_i(0)+\Pr_i(1)\big)\big[1-\Pr_{i,i-k}(0\mid 1) - \Pr_{i,i-k}(1\mid 0)\big], \\ = \quad & 1-\big(\Pr_{i,i-k}(0\mid 1)+\Pr_{i,i-k}(1\mid 0)\big).\end{aligned}$$

which, according to our model established in Def. 5.2, provides as announced:

$$\rho_k = 1-(1-\delta_k) = \delta_k.$$

□

Thus, our model for the correlation anomaly seems a even more validated by the fact that its significant parameter is exactly equal to the theoretical autocorrelation of the process it describes.

### 5.3. Constraints between the parameters of the model

We already saw thanks to Prop. 5.1 that the frequency anomaly was linked to the correlation anomaly, and more specifically that the probabilities $\Pr_i(x)$ and $\Pr_{i,i-k}\big(x \mid y\big)$, $(x,y) \in \{0,1\}^2$ are constrained by one another. Adding the constraint that these are probabilities, and therefore have values in $[0,1]$, we obtain the following property on the values of $\xi$ and $\delta_k$:

PROPERTY 5.4. *Let $\xi$ and $\delta_k$ be the terms derived from the models defined in Def. 4.1 and 5.2. These terms are linked by the following constraint:*

$$\max\left(-\frac{1-\xi}{1+\xi}, -\frac{1+\xi}{1-\xi}\right) \leqslant \delta \leqslant 1.$$

PROOF. As a reminder, Prop. 5.1 provides the following constraint:

$$\Pr_i(1) \times \Pr_{i,i-k}(0\mid 1) = \Pr_i(0) \times \Pr_{i,i-k}(1\mid 0).$$

$\Pr_{i,i-k}(0 \mid 1)$ and $\Pr_{i,i-k}(1 \mid 0)$ being probabilities, both must be with values in $[0,1]$.

Thus $0 \leqslant \Pr_{i,i-k}(0 \mid 1) \leqslant 1$ provides:

$$0 \leqslant \Pr_{i,i-k}(1 \mid 0) \leqslant \min\left(1, \frac{1+\xi}{1-\xi}\right),$$

and $0 \leqslant \Pr_{i,i-k}(1 \mid 0) \leqslant 1$ provides:

$$0 \leqslant \Pr_{i,i-k}(0 \mid 1) \leqslant \min\left(1, \frac{1-\xi}{1+\xi}\right).$$

The first inequation for example leads to:

$$0 \leqslant \frac{(1+\xi)(1-\delta_k)}{2} \leqslant \min\left(1, \frac{1+\xi}{1-\xi}\right),$$

which translates into:

$$0 \leqslant 1-\delta_k \leqslant \min\left(\frac{2}{1+\xi}, \frac{2}{1-\xi}\right),$$

or:

$$\max\left(-\frac{1-\xi}{1+\xi}, -\frac{1+\xi}{1-\xi}\right) \leqslant \delta_k \leqslant 1.$$

The second inequation then provides the exact same inequation, which demonstrates the property. □

Thus, although the models have been defined for $(\xi, \delta_k) \in ]-1,1[$, it appears that these two parameters cannot evolve in the whole space independently form one another, but that they evolve in the space depicted in Fig. 15.

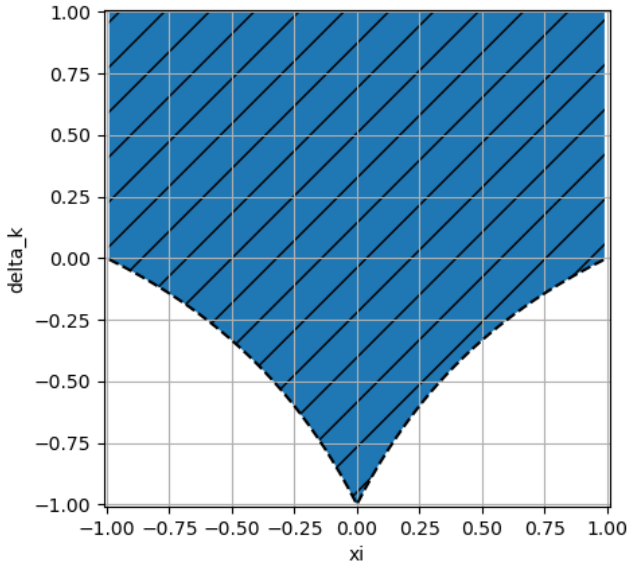



FIGURE 15. Pairs $(\xi, \delta_k)$ for which the model for the correlation anomaly is defined

Intuitively, this constraint is natural, as $\delta_k \to -1$ means that bits distant from $k > 0$ will tend to take complementary values. And the more this phenomenon is accentuated, the greater the tendency for the overall proportion of 0 and 1 bits to tend towards 50%, which translates into $\xi \to 0$. Conversely, $\delta_k > 0$ means that bits distant from $k > 0$ will have tend to take on identical values, which has no impact on the global proportion of bits 0 and 1. For example even in the extreme case of a sequence composed of two subsequences of identical values 000000...00000011111...111111, then $\delta_k$ will be worth almost 1 for any $k > 0$, while the value of $\xi$ will depend entirely on where in the sequence the inversion of values takes place. This therefore illustrates why $\xi$ is constrained when $\delta_k$ is negative but not when $\delta_k$ is positive.

As a conclusion, we have established a model for correlations between bits distant from $k > 0$ in a sequence, and we have proven that the significant parameter of the model, noted $\delta_k$, is identically equal to the theoretical autocorrelation of the stochastic process modeling the bits of the sequence. The model we propose therefore appears to be optimal for studying correlations, and we now need to find a statistic to estimate the value of the parameter $\delta_k$ in order to establish an optimal test in the sense of our Def. 3.15.

CHAPTER 6

# Overlap between the Monobit and the Autocorrelation test (AIS 20/31)

As we mentioned in subsection 3.3.4, the Autocorrelation test (T5) from the AIS 20/31 is impacted by the frequency anomaly, for which we have proven that the Monobit test (T1) is optimal, apart from one linear transformation. We now seek to formalize the interaction between the two tests.

## 6.1. Theoretical approach from Lubicz

In [**51**], Lubicz defines a notion of dependency between finite statistical tests. Again, this definition appears to apply to statistics rather than tests, and in the remainder of this subsection, we will then present Lubicz's results as being about statistics rather than tests.

DEFINITION 6.1 (Dependency between statistics [**51**]). *For $F$ and $F'$ two statistics, represented by their respective automata (see subsection 3.2.6 for more details):*

- *If the set of states of the product automaton $F \times F'$ is exactly the Cartesian product of the states of $F$ and $F'$, then the two statistics are completely independent.*
- *If $F \times F' = F$, then the two statistics have maximum dependency (and there is a morphism from $F$ to $F'$).*

*More generally, the larger the product automata of $F$ and $F'$ (so the more accessible states it contains), the more independent the statistics.*

It is then possible to show that, according to this definition, the Monobit and Autocorrelation statistics of the AIS 20/31 have maximum dependency. In fact, taking for example, the case of the autocorrelation between directly successive bits, studied by statistic $\mathscr{A}_1$, the automaton produced by the two statistics is shown in Fig. 16.

The states are labeled with the convention '$b$' × '$xy$', with $(b,x,y) \in \{0,1\}$, where '$b$' is the label of the states of the automaton of the Monobit statistic and $'xy'$ is the label of the states of the Autocorrelation statistic ($\mathscr{A}_1$). As $'b'$ and $'x'$ both represent the last bit read, the states for which '$b$' ≠ '$x$' are naturally inaccessible, and it appears that the product automaton of the two statistics is therefore exactly the automaton of the Monobit statistic.

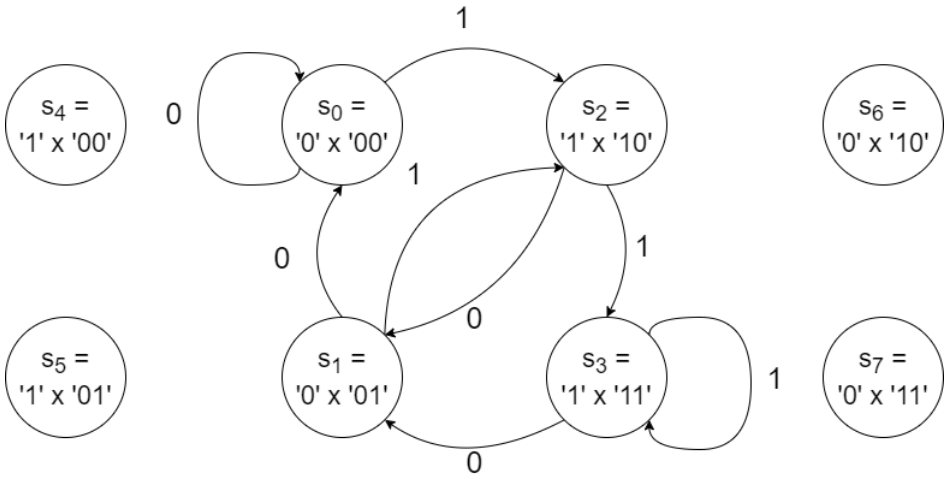



FIGURE 16. Product automaton of the automata of the Monobit and Autocorrelation statistics.

A strictly identical reasoning can be performed to show that, for any $k > 0$, the automaton produced by the Monobit and Autocorrelation ($\mathscr{A}_k$) statistics will be perfectly equal to the automaton of the Autocorrelation statistic ($\mathscr{A}_k$). Thus, the two statistics (and the tests they are based on) have maximum dependency according to Lubicz's definition.

> *Remark :* By reusing the labeling introduced earlier for the states of the automata of the Monobit and Autocorrelation ($\mathscr{A}_k$) statistics, the morphism $\chi_k$ to transform the automaton of the Autocorrelation statistic into the automaton of the Monobit statistic is the following:
>
> $$\begin{array}{rccl} \chi_k : & \{s_0, \ldots, s_{2^{k+1}-1}\} & \longrightarrow & \{s'_0, s'_1\} \\ & s_i = {}'b\ b_k \ldots b'_1 & \longmapsto & \begin{cases} s'_0 = {}'0' & \text{if } b = {}'0', \\ s'_1 = {}'1' & \text{if } b = {}'1'. \end{cases} \end{array}$$
>
> In other words, for any $k > 0$, the automaton of the Autocorrelation statistic ($\mathscr{A}_k$) can be reduced into the automaton of the Monobit statistic by gathering the states whose labels have the same most significant bit. For example, the reduction of the automaton of the Autocorrelation statistic ($\mathscr{A}_1$) is depicted in Fig. 17.

## 6.2. A new approach to overlap between statistical tests

Although the approach proposed by Lubicz allows for a formal demonstration of the fact that the Monobit test and the Autocorrelation test are functionally linked (due to their statistics

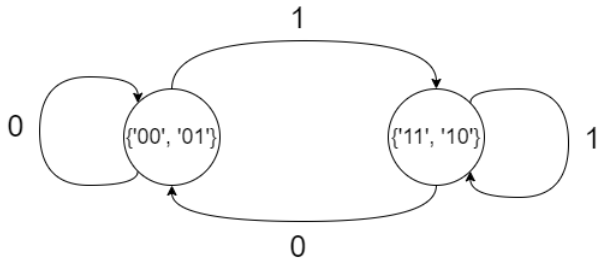



FIGURE 17. Reduction of the automaton of the Autocorrelation test ($\mathscr{A}_1$) into the automaton of the Monobit test.

being fully dependent), we propose another, more practical approach of the concept of interaction, or **overlap**, between statistical tests. Using the terms we used in our definition of a statistical test (Def. 3.7), we define the overlap between two tests as follows:

DEFINITION 6.2 (Overlap between statistical tests). *Let $X, X'$ be two statistics, and $T_{X,S_0}$, $T_{X',S'_0}$ be statistical tests defined from these statistics. We say that $T_{X,S_0}$ and $T_{X',S'_0}$* overlap *with one another if and only if $\mathbb{E}[X]$ and $\mathbb{E}\left[X'\right]$ are function of one or several common parameters.*

THEOREM 6.3. *The Autocorrelation test from the AIS 20/31 overlaps with the Monobit test.*

PROOF. With the models of frequency and correlation anomalies as defined in Def. 4.1 and 5.2, the expectation of the Monobit statistic $M$ for a sequence composed of $N$ bits is equal to:

$$\mathbb{E}[M] = N \times \frac{1+\xi}{2}$$

And the expected value of the autocorrelation statistic $\mathscr{A}_k$ is equal to:

$$\mathbb{E}[\mathscr{A}_k] = \mathbb{E}\left[\sum_{i=k}^{N} b_i \oplus b_{i-k}\right]$$

where the sequence $(b_i)_{1\leqslant i\leqslant N}$ is seen as the sequence of realizations of the stochastic process $\{B_i\}$. In this regard, the expected value of $\mathscr{A}_k$ is then expressed as such:

$$\begin{aligned}
\mathbb{E}[\mathscr{A}_k] &= N \times \Big[\Pr_{i,i-k}(0,1) + \Pr_{i,i-k}(1,0)\Big], \\
&= N \times \Big[\Pr_{i,i-k}(0 \mid 1) \times \Pr_{i-k}(1) + \Pr_{i,i-k}(1 \mid 0)\Pr_{i-k}(0)\Big], \\
&= N \times \left[\frac{(1-\xi)(1-\delta_k)}{2} \times \frac{1+\xi}{2} + \frac{(1+\xi)(1-\delta_k)}{2} \times \frac{1-\xi}{2}\right], \\
&= N \times \frac{(1-\xi^2)(1-\delta_k)}{2}.
\end{aligned}$$

Thus, the expected value of the Monobit and of the Autocorrelation statistics are both functions of the parameter $\xi$. The two tests therefore overlap.

□

This expression of the expected value of $\mathscr{A}_k$ also makes it possible to visualize the practical impact of an global disproportion of bits 0 and 1 on the test, which will be less than that of a real autocorrelation between the bits, in the sense that, at an equal amplitude, an autocorrelation will cause a greater deviation of the value of $\mathscr{A}_k$ (because $\xi \in ]-1,1[$).

It then seems interesting to consider the design of a statistical test that would be only impacted by a correlation anomaly. Ideally, we are looking for a statistic that is an unbiased estimator of the parameter $\delta_k$, so that we can derive an optimal test (in the sense of our Def. 3.15).

CHAPTER 7

# Enhanced autocorrelation statistic

## Contents



From the proof of the theorem 5.3, it appears that a new test, which would be optimal with regard to the correlation anomaly can easily be designed from the expression of the conditional probabilities established in Def. 5.2. Indeed, as a reminder, our model provides the following equation:

$$1 - \left(\Pr_{i,i-k}(0 \mid 1) + \Pr_{i,i-k}(1 \mid 0)\right) = \delta_k.$$

> *Remark:* In the latest draft of the AIS 20/31, the value $1 - \left(\Pr_{i,i-k}(0 \mid 1) + \Pr_{i,i-k}(1 \mid 0)\right)$ is used in the security functional requirement PTG.2.2 ([**67**], (279) p.45) with $k = 1$ and $k = 2$, and is described as the "*1-step and 2-step dependencies*" respectively. And although we proved in Th. 5.3 that this value is indeed equal to the theoretical autocorrelation of a stationary binary-valued stochastic process, the authors of the standard chose not to develop a statistic based on it for their autocorrelation test.

### 7.1. Definition of the enhanced autocorrelation statistic

Using our model for the correlation anomaly, we then propose a new statistic for the study of correlations between the bits of a sequence.

DEFINITION 7.1 (Enhanced autocorrelation statistic). *For a binary sequence $(b_i)_{1 \leqslant i \leqslant N}$, $N \in \mathbb{N}$, seen as a sequence of realizations of the process $\{B_i\}$ described by the model established in Def. 5.2, the* ***enhanced autocorrelation statistic*** *$\mathscr{A}_k^\star$ is defined as:*

$$\mathscr{A}_k^\star = 1 - \left(\frac{N_{10}^k}{N_1} + \frac{N_{01}^k}{N_0}\right).$$

*where $N_x$ is the number of occurrences of the bit $x$ and $N_{xy}^k$ is the number of occurrences of the*

*pair* $(b_{i-k} = x, b_i = y), (x, y) \in \{0,1\}^2$ *in the sequence.*

THEOREM 7.2. *Let* $\delta_k$ *be the term characterizing the correlations in the model introduced in Def. 5.2. The statistic* $\mathscr{A}_k^\star$ *defined above is an* ***unbiased estimator of*** $\delta_k$.

PROOF. Let $N_0$, $N_1$, $N_{01}^k$ and $N_{10}^k$ the terms introduced in Def. 7.1, and $N \in \mathbb{N}, N > 0$ the length of the studied sequence. Then, by construction:

$$\mathbb{E}\left[\mathscr{A}_k^\star\right] = 1 - \mathbb{E}\left[\frac{N_{10}^k}{N_1}\right] - \mathbb{E}\left[\frac{N_{01}^k}{N_0}\right].$$

We will first prove that:

$$\mathbb{E}\left[\frac{N_{10}^k}{N_1}\right] = \Pr_{i,i-k}(0 \mid 1).$$

As our model is defined for $\xi \notin \{-1, 1\}$, we have $1 \leqslant N_1 \leqslant N-1$. Then, in all generality:

$$\begin{aligned}\mathbb{E}\left[\frac{N_{10}^k}{N_1}\right] &= \sum_{n_1=1}^{N-1}\sum_{n_{10}=1}^{n_1}\frac{n_{10}}{n_1}\Pr\left(N_{10}^k = n_{10}, N_1 = n_1\right),\\ &= \sum_{n_1=1}^{N-1}\Pr(N_1 = n_1) \times \sum_{n_{10}=1}^{n_1}\frac{n_{10}}{n_1}\Pr\left(N_{10}^k = n_{10} \mid N_1 = n_1\right).\end{aligned}$$

$N_{10}^k$ represents the number of pairs $(b_{i-k} = 1, b_i = 0)$, so $\Pr\left(N_{10}^k = n_{10} \mid N_1 = n_1\right)$ is exactly the probability of having $n_{10}$ occurrences of the bit 0 after one of the $n_1$ bits 1. The random variable $N_{10}^k \mid N_1 = n_1$ therefore follows a binomial distribution $\mathscr{B}(n_1, \Pr_{i,i-k}(0 \mid 1))$. Thus:

$$\begin{aligned}\mathbb{E}\left[\frac{N_{10}^k}{N_1}\right] &= \sum_{n_1=1}^{N-1}\Pr(N_1 = n_1) \times \sum_{n_{10}=1}^{n_1}\frac{n_{10}}{n_1}\binom{n_1}{n_{10}} \times \Pr_{i,i-k}(0 \mid 1)^{n_{10}}\left(1 - \Pr_{i,i-k}(0 \mid 1)\right)^{n_1 - n_{10}},\\ &= \sum_{n_1=1}^{N-1}\Pr(N_1 = n_1) \times \sum_{n_{10}=1}^{n_1}\binom{n_1 - 1}{n_{10} - 1} \times \Pr_{i,i-k}(0 \mid 1)^{n_{10}}\left(1 - \Pr_{i,i-k}(0 \mid 1)\right)^{n_1 - n_{10}},\\ &= \Pr_{i,i-k}(0 \mid 1) \times \sum_{n_1=1}^{N-1}\Pr(N_1 = n_1),\\ &= \Pr_{i,i-k}(0 \mid 1).\end{aligned}$$

With a rigorously analogous reasoning, we also prove that:

$$\mathbb{E}\left[\frac{N_{01}^k}{N_0}\right] = \Pr_{i,i-k}(1 \mid 0).$$

Therefore, as announced:

$$\begin{aligned}\mathbb{E}\left[\mathscr{A}_k^\star\right] &= 1 - \Pr_{i,i-k}(0 \mid 1) - \Pr_{i,i-k}(1 \mid 0),\\ &= \delta_k.\end{aligned}$$

□

As a reminder, $\delta_k$ is an image of the theoretical autocorrelation of the process $\{B_i\}$ according to our model (see Th. 5.3). Thus, just as the statistic $\hat{\xi}$ made it possible to define an optimal test for the frequency anomaly, the enhanced autocorrelation statistic $\mathscr{A}_k^\star$ allows for the definition of an **optimal test** (as per Def. 3.15) **for the correlation anomaly** as modeled in Def. 5.2.

> *Remark:* To be coherent with the Autocorrelation test from the AIS 20/31, we could define an enhanced autocorrelation test by applying our statistic to sequences of 20,000 bits, and allowing for a deviation of 6.96% from the ideal value of 0. The accepted range of the test would then be $S_0 = [-0.696, 0.696]$.

Moreover, this statistic does not suffer from any overlap (as per Def. 6.2) with the Monobit statistic, since $\xi$ is absent from the expected value of $\mathscr{A}_k^\star$.

However, in the sense of the definition of dependency between tests proposed by Lubicz [**51**], the Monobit test and a test based on our statistic $\mathscr{A}_k^\star$ still present a maximal dependency. Indeed, the automaton representing the application of the statistic $\mathscr{A}_k^\star$ is the same automaton that is used to apply the Autocorrelation statistic $\mathscr{A}_k$ from the AIS 20/31 (Fig. 14 for the case k = 1). The statistic $\mathscr{A}_k^\star$ is then computed by counting the number of passes in all states whose labels' most significant bits are 0 for $N_0$ and 1 for $N_1$, as well as that the number of passes in the states for which the most and least significant bits in the labels are respectively 0 and 1 for $N_{10}^k$ and 1 and 0 for $N_{01}^k$.

While a contradiction seems to appear between our definition and the definition of Lubicz, we will see in the next section that the influence of the frequency anomaly (characterized by the statistic $\hat{\xi}$, or, equivalently, by the Monobit statistic) is actually found in the variance of $\mathscr{A}_k^\star$.

## 7.2. Variance of the enhanced autocorrelation statistic

Similarly to the statistic $\hat{\xi}$ established in Def. 4.2, although the expected value of $\mathscr{A}_k^\star$ independent of $\xi$, it appears that its variance is not. However, unlike the variance of $\hat{\xi}$, there is no analytical expression for the variance of $\mathscr{A}_k^\star$. More precisely, we have the following property:

PROPERTY 7.3 (Variance of the statistic $\mathscr{A}_k^\star$). *Let, $k > 0$, $\mathscr{A}_k^\star$ be the enhanced autocorrelation statistic (Def. 7.1), and $N_0$, $N_1$ be the random variables describing the number of bits* 0 *and* 1 *in the tested sequence. We then have:*

$$\begin{aligned}\mathrm{Var}\left(\mathscr{A}_k^\star\right) = \quad & \Pr_{i,i-k}(0 \mid 1)\,(1-\Pr_{i,i-k}(0 \mid 1)) \times \mathbb{E}\left[\frac{1}{N_1}\right] \\ & + \Pr_{i,i-k}(1 \mid 0)\,(1-\Pr_{i,i-k}(1 \mid 0)) \times \mathbb{E}\left[\frac{1}{N_0}\right].\end{aligned}$$

PROOF. For a sequence of $N$ bits, we have, in all generality:

$$\begin{aligned}\mathrm{Var}\left(\mathscr{A}_k^\star\right) &= \mathrm{Var}\left(\frac{N_{10}^k}{N_1}+\frac{N_{01}^k}{N_0}\right), \\ &= \mathbb{E}\left[\left(\frac{N_{10}^k}{N_1}\right)^2\right] + 2\times\mathbb{E}\left[\frac{N_{10}^k N_{01}^k}{N_1 N_0}\right] + \mathbb{E}\left[\left(\frac{N_{01}^k}{N_0}\right)^2\right] - \left(\mathbb{E}\left[\frac{N_{10}^k}{N_1}+\frac{N_{01}^k}{N_0}\right]\right)^2.\end{aligned}$$

We have already shown in the proof of Th. 7.2 that:

$$\left(\mathbb{E}\left[\frac{N_{10}^k}{N_1}+\frac{N_{01}^k}{N_0}\right]\right)^2 = (1-\delta_k)^2.$$

In a similar fashion to the proof of the previous theorem, we can find an expression of the "product" expected value:

$$\begin{aligned}\mathbb{E}\left[\frac{N_{10}^k N_{01}^k}{N_1 N_0}\right] &= \sum_{n_1=1}^{N-1} P(N_1=n_1) \times \sum_{n_{10}=1}^{n_1}\sum_{n_{01}=1}^{N-n_1} \frac{n_{10}n_{01}}{n_1(N-n_1)} \\ &\quad \times \binom{n_1}{n_{10}} \Pr_{i,i-k}(0 \mid 1)^{n_{10}} (1-\Pr_{i,i-k}(0 \mid 1))^{n_1-n_{10}} \\ &\quad \times \binom{N-n_1}{n_{01}} \Pr_{i,i-k}(1 \mid 0)^{n_{01}} (1-\Pr_{i,i-k}(1 \mid 0))^{N-n_1-n_{01}}, \\ &= \Pr_{i,i-k}(0 \mid 1) \times \Pr_{i,i-k}(1 \mid 0). \\ &= \frac{(1-\xi^2)(1-\delta_k)^2}{4}.\end{aligned}$$

Then, we proceed similarly to find the expression of the other expected values:

$$
\begin{aligned}
\mathbb{E}\left[\left(\frac{N_{10}^k}{N_1}\right)^2\right] &= \sum_{n_1=1}^{N-1}\sum_{n_{10}=1}^{n_1}\frac{n_{10}^2}{n_1^2}\Pr\left(N_{10}^k = n_{10} \mid N_1 = n_1\right) \times \Pr\left(N_1 = n_1\right), \\
&= \sum_{n_1=1}^{N-1}\Pr\left(N_1 = n_1\right) \times \sum_{n_{10}=1}^{n_1}\frac{n_{10}^2}{n_1^2}\binom{n_1}{n_{10}}\Pr_{i,i-k}(0 \mid 1)^{n_{10}}\left(1-\Pr_{i,i-k}(0 \mid 1)\right)^{n_1-n_{10}}, \\
&= \sum_{n_1=1}^{N-1}\Pr\left(N_1 = n_1\right) \times \sum_{n_{10}=1}^{n_1}\frac{n_{10}}{n_1}\binom{n_1-1}{n_{10}-1}\Pr_{i,i-k}(0 \mid 1)^{n_{10}}\left(1-\Pr_{i,i-k}(0 \mid 1)\right)^{n_1-n_{10}}, \\
&= \sum_{n_1=1}^{N-1}\Pr\left(N_1 = n_1\right) \times \frac{\Pr_{i,i-k}(0 \mid 1)}{n_1} \times \left((n_1-1)\Pr_{i,i-k}(0 \mid 1)+1\right), \\
&= \Pr_{i,i-k}(0 \mid 1)^2 \times \sum_{n_1=1}^{N-1}\Pr\left(N_1 = n_1\right) \\
&+ \Pr_{i,i-k}(0 \mid 1)\left(1-\Pr_{i,i-k}(0 \mid 1)\right) \times \sum_{n_1=1}^{N-1}\frac{1}{n_1}\Pr\left(N_1 = n_1\right), \\
&= \Pr_{i,i-k}(0 \mid 1)^2 + \Pr_{i,i-k}(0 \mid 1)\left(1-\Pr_{i,i-k}(0 \mid 1)\right) \times \sum_{n_1=1}^{N-1}\frac{1}{n_1}\Pr\left(N_1 = n_1\right), \\
&= \Pr_{i,i-k}(0 \mid 1)^2 + \Pr_{i,i-k}(0 \mid 1)\left(1-\Pr_{i,i-k}(0 \mid 1)\right) \times \mathbb{E}\left[\frac{1}{N_1}\right].
\end{aligned}
$$

And:

$$
\begin{aligned}
\mathbb{E}\left[\left(\frac{N_{01}^k}{N_0}\right)^2\right] &= \Pr_{i,i-k}(1 \mid 0)^2 + \Pr_{i,i-k}(1 \mid 0)\left(1-\Pr_{i,i-k}(1 \mid 0)\right) \times \sum_{n_0=1}^{N-1}\frac{1}{n_0}\Pr\left(N_0 = n_0\right), \\
&= \Pr_{i,i-k}(1 \mid 0)^2 + \Pr_{i,i-k}(1 \mid 0)\left(1-\Pr_{i,i-k}(1 \mid 0)\right) \times \mathbb{E}\left[\frac{1}{N_0}\right].
\end{aligned}
$$

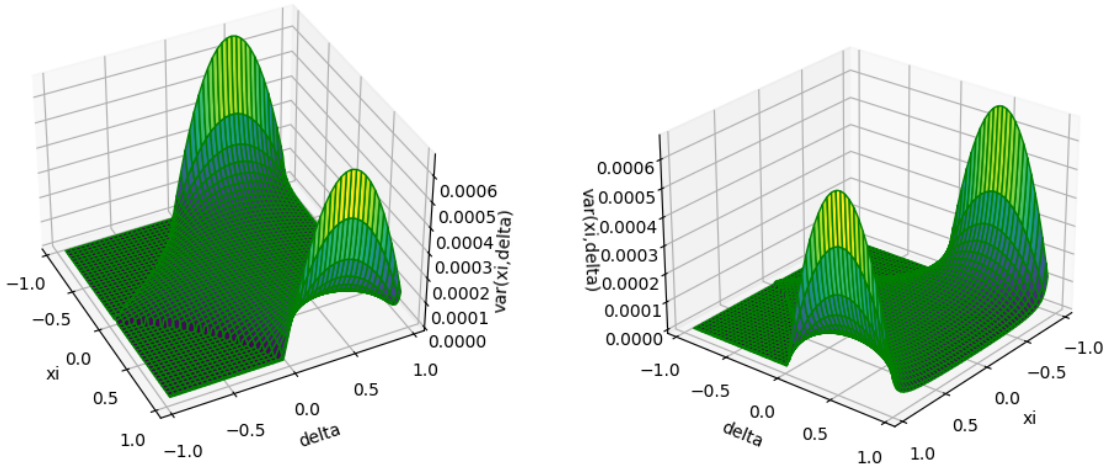
We can thus simplify the expression of the variance by noticing that:

$$
P(0 \mid 1)^2 + P(1 \mid 0)^2 = \frac{(1+\xi^2)(1-\delta_k)^2}{2} = \left(\mathbb{E}\left[\frac{N_{10}^k}{N_1} + \frac{N_{01}}{N_0}\right]\right)^2 - 2 \times \mathbb{E}\left[\frac{N_{10}^k N_{01}^k}{N_1 N_0}\right],
$$

which provides in the end:

$$
\begin{aligned}
\operatorname{Var}\left(\mathscr{A}_k^\star\right) = \quad & \Pr_{i,i-k}(0 \mid 1)\left(1-\Pr_{i,i-k}(0 \mid 1)\right) \times \mathbb{E}\left[\frac{1}{N_1}\right] \\
& + \Pr_{i,i-k}(1 \mid 0)\left(1-\Pr_{i,i-k}(1 \mid 0)\right) \times \mathbb{E}\left[\frac{1}{N_0}\right].
\end{aligned}
$$

□




(A) Perspective view (B) Perspective view (bis)

FIGURE 18. Numerical computation of $\mathrm{Var}\left(\mathscr{A}_k^\star\right)$, for $N = 15\,000$

In the same way that $N_{10}^k$ and $N_{01}^k$ follow binomial laws, the random variables $N_0$ and $N_1$ follow laws $\mathscr{B}(N, \Pr_i(0))$ and $\mathscr{B}(N, \Pr_i(1))$ respectively, which enables us to simulate the variance of $\mathscr{A}_k^\star$ when $\xi$ and $\delta_k$ vary in the space defined in Prop. 5.4. Figure 18 depicts the numerical computation of this variance, for $(\xi, \delta_k) \in [-0.95, 0.95]^2$ and $N = 15{,}000$.

The first observation we can make is that this variance is perfectly symmetrical in $\xi$, which was predictable and reinforces the idea that bits 0 and 1 are at completely interchangeable in our model. The variance is also roughly symmetrical in $\delta_k$ for low values of $|\xi|$ (namely for $\xi \in [-0.1, 0.1]$) and minimal when $\delta_k$ is close to $-1$ and 1. Thus, the larger the anomaly, the more precise its characterization with our statistic will be.

On the other hand, the limit of the enhanced autocorrelation statistic lies in the case where $\xi$ is very close to either $-1$ or 1. When there are too many bits of identical value, it becomes difficult to distinguish a global frequency anomaly from a real correlation anomaly between bits distant of $k > 0$. Mathematically, this peak in the variance is explained by Jensen's inequation [**41**]. Since the function $x \mapsto \frac{1}{x}$ is convex, Jensen inequation provides:

$$\mathbb{E}\left[\frac{1}{N_1}\right] \geqslant \frac{1}{\mathbb{E}[N_1]} = \frac{1}{N \times \Pr_i(1)} = \frac{2}{N \times (1+\xi)} \underset{\xi \to -1}{\to} +\infty$$

and similarly:

$$\mathbb{E}\left[\frac{1}{N_0}\right] \geqslant \frac{1}{\mathbb{E}[N_0]} = \frac{1}{N \times \Pr_i(0)} = \frac{2}{N \times (1-\xi)} \underset{\xi \to +1}{\to} +\infty$$

*Remark :* The flat areas on Fig. 18 are the pairs $(\xi, \delta_k)$ for which the model is not defined. We have chosen to set the variance to 0 in these areas to not impair the readability.

## 7.3. Applying the enhanced autocorrelation statistic on simulated sequences

Several simulations have been carried out on different sequences to assess the ability of the statistic $\mathscr{A}_k^\star$ provided in Def. 5.2 to characterize the presence of correlation between the bits of the sequence.

In subsection 7.3.1, we will look at the `RDRAND` and `RDSEED` functions from Intel's random number generation library [**40**] (the latter being used to provide seeds for pseudo-random generators). These functions have been thoroughly analyzed by Shrimpton and Terashima [**74**], who proved in particular that the RDRAND function is very robust. We will use `RDRAND` as a reference to test the statistic $\mathscr{A}_k^\star$, and our implementation of it, by verifying that when no correlation anomaly is present (as should be the case with `RDRAND`), $\mathscr{A}_k^\star = 0$ for any $k > 0$.

Secondly, in subsection 7.3.2, we will study biased random sequences with known frequency and correlation anomalies to demonstrate the usefulness of our enhanced autocorrelation statistic compared to the AIS 20/31 autocorrelation statistic. To this end, we will compute $\mathscr{A}_k$ and $\mathscr{A}_k^\star$ for $1 \leqslant k \leqslant 128$ on those sequences, and compare the graphs obtained with both statistics. Having access to the values of the anomaly parameters $\xi$ and $\delta_k$ for each sequence we generated also enabled us to further verify our implementation of $\mathscr{A}_k^\star$.

Finally, subsection 7.3.3 will focus on sequences generated by an implementation on an FPGA-type reprogrammable circuit of the Elementary TRNG as defined by Fischer and Lubicz in [**29**], and compute $\mathscr{A}_k$ and $\mathscr{A}_k^\star$ for $1 \leqslant k \leqslant 128$ on those sequences, similarly to subsection 7.3.2.

### 7.3.1. `RDRAND` and `RDSEED`.

For both `RDRAND` and `RDSEED`, we generated sequences of bits according to Alg. 4, which we implemented in `C` (replacing `RDRAND_32()` by `RDSEED_32()` to generate sequences with the second function).

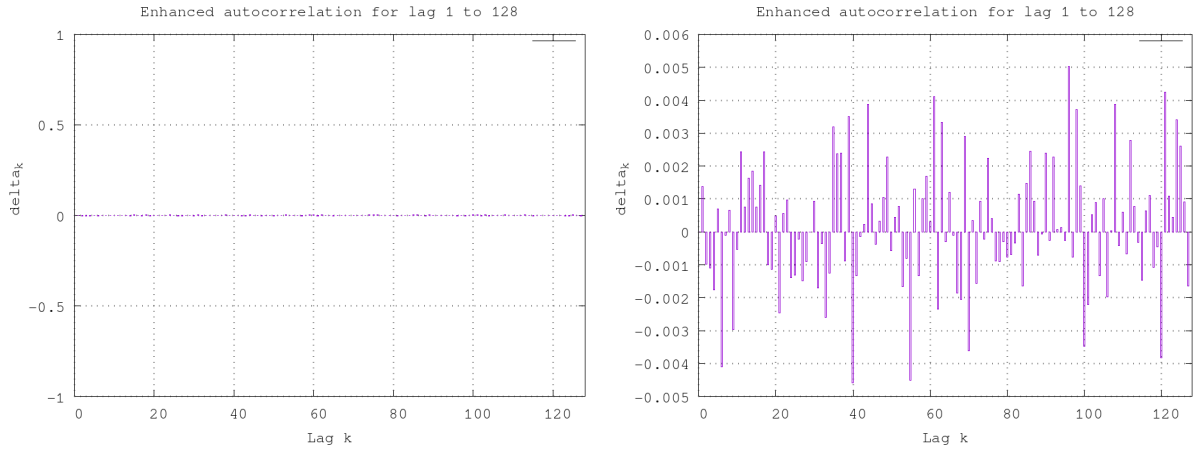
The processor used is a 14nm Intel Xeon E5-2630 v4 (Broadwell architecture, 5th generation). The random data generated by `RDSEED` comes from a hardware entropy source (taking advantage of thermal noise in silicon circuits), which feeds an AES-CBC-MAC based conditioner. The data generated by `RDRAND` comes from the same system, to which an AES-CTR hardware block is added to eliminate residual dependencies on the cryptographic block (see [**40**] §3.2).

7.3.1.1. *`RDRAND`.* Given the way the function is built, and from Shrimpton and Terashima's work, the sequences generated by `RDRAND` are expected to be of very good quality (in terms of randomness), which would translate, in our case, into null dependency coefficients $\delta_k$ for any

**Algorithm 4** Generation of random bit sequences with `RDRAND`

**Input:** $N_{ints} > 0$.
**Output:** Sequence of $32 \times N_{ints}$ random bits.

```
RandSeq ← []
for 0 ⩽ i < N_ints do
    rand_int ← RDRAND_32()          ▷ RDRAND_32() returns a random 32 bits integer.
    for 0 ⩽ j < 32 do
        RandSeq[32 × i + j] ← rand_int % 2
        rand_int = rand_int / 2     ▷ / is the integer division operator.
    end for
end for
```




(A) Enhanced autocorrelation statistic (B) Zoomed-in view

FIGURE 19. Application of $\mathscr{A}_k^\star$ to sequences generated with `RDRAND`, 320,000 bits

lag $k > 0$.

The two graphs of Fig. 19 depict the application of our statistic $\mathscr{A}_k^\star$ (also implemented in `C`) on a 320,000-bit sequence. Both graphs represent the amplitude of the coefficients $\delta_k$ for every $k > 0$ between 1 and 128, the graph on the right simply being a view adjusted to the amplitude of the different $\delta_k$ while the graph on the left is a fixed global view where the axis for $\delta_k$ varies between $-1$ and 1.

As anticipated, the various coefficients $\delta_k$ are all almost null, which indicates that, from the point of view of correlations, the `RDRAND` function is of very good quality (no significant correlation) even when used in the most "basic" way of simply concatenating every output integer.

7.3.1.2. *RDSEED*. As the RDSEED function has been designed to supply seeds to pseudo-random generators, the numbers it generates do not need to be of as high quality as those generated by RDRAND. But the study of a sequence of 320,000 bits generated by RDSEED yielded results strictly analogous to those obtained with RDRAND, both in the shape of the graphs and in their amplitude. It therefore appears that the randomness provided before the AES post-processing does not show any significant correlation either.

**7.3.2. Simulated sequences with known frequency and correlation anomalies.** For the following simulations, sequences of 100,000 bits were generated with a fixed frequency and correlation anomaly between bits distant of $k > 0$. To do this, we used the RDRAND function to generate sequences according to Alg. 5, also implemented in C. See Def. 5.2 for the expression of $\Pr_{j,j-k}\left(\overline{b_{j-k}} \mid b_{j-k}\right)$ as a function of $\xi$ and $\delta_k$.

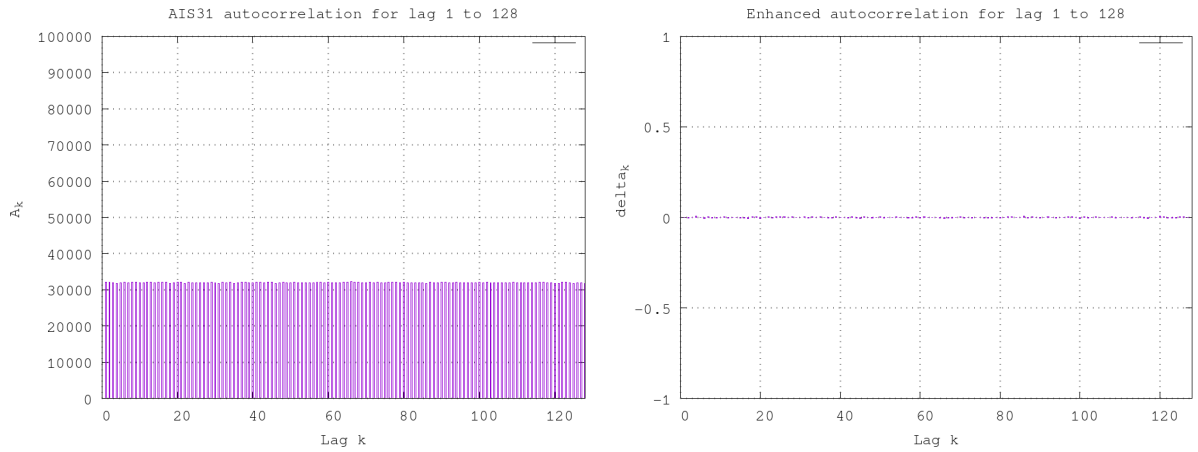
**Algorithm 5** Generation of binary sequences with fixed anomaly parameters $\xi$ and $\delta_k$

**Input:** $N > 0, N > k > 0, (\xi, \delta_k) \in ]-1,1[^2$.
**Output:** Sequence of $N$ bits with anomaly characteristics $\xi$ and $\delta_k$.

```
RandSeq ← []
for 0 ⩽ i < k do                          ▷ We first generate k random bits with RDRAND
    RandSeq[i] ← RDRAND({0,1})            ▷ RDRAND({0,1}) returns 0 or 1 with equal prob.
end for
for k ⩽ j < N do
    decision ← RDRAND([0,1])   ▷ RDRAND([0,1]) returns a real number in [0,1] with uniform probability.
    b_{j-k} ← RandSeq[j − k]
    if decision < Pr_{j,j-k}(¬b_{j-k} | b_{j-k}) then
        RandSeq[j] ← ¬b_{j-k}
    else
        RandSeq[j] ← b_{j-k}
    end if
end for
```

To illustrate that the enhanced autocorrelation statistic is at least as efficient as the autocorrelation statistic of the AIS 20/31, and that it is not affected by a frequency anomaly, i.e. by a simple overall disproportion of bits 0 and 1, we then generated three sequences. As a reminder, in the ideal case, the values of the autocorrelation statistics of AIS 20/31 $\mathscr{A}_k$ and our enhanced autocorrelation statistic $\mathscr{A}_k^\star$ are : $\mathscr{A}_k = \frac{N}{2}$ and $\mathscr{A}_k^\star = 0$, for all $k > 0$. Any significant deviation from these values in the graphs in Fig. 20, 21 and 22 will therefore mean that an anomaly has been detected.

The first sequence has been generated with parameters $\xi = 0.6$ and $\delta_k = 0$ for all $k > 0$, meaning that it suffers from a frequency anomaly, but from no correlation anomaly. The analysis of




(A) Autocorrelation statistic of the AIS 20/31 (B) Enhanced autocorrelation statistic

FIGURE 20. Autocorrelation statistics $\mathscr{A}_k$ vs $\mathscr{A}_k^\star$, $N = 100{,}000$, $\xi = 0.6$, $\delta_k = 0$ for all $k > 0$

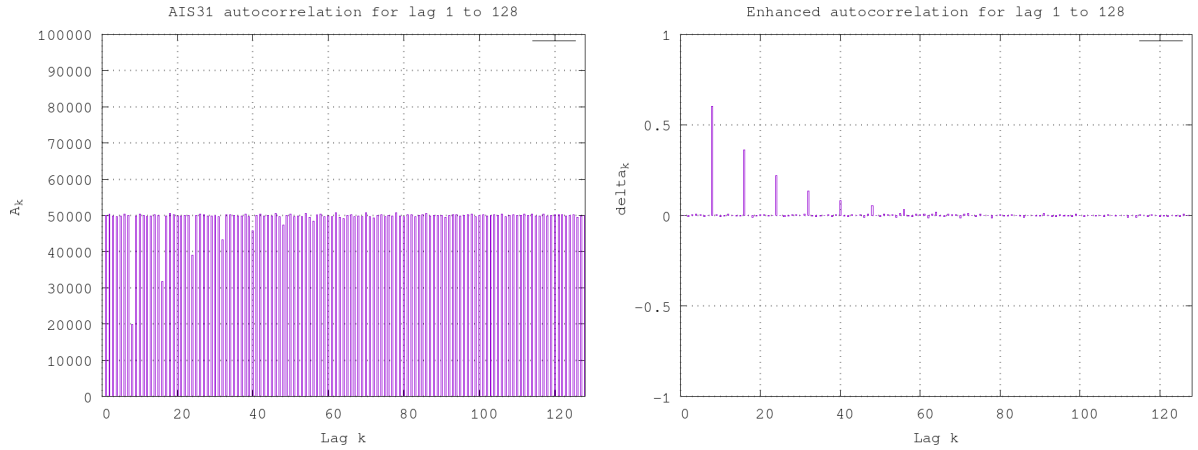
the sequence then leads to the graphs in Fig. 20.

As expected, the graphs show that when a sequence is only affected by a frequency anomaly, the Autocorrelation test from AIS 20/31 test can fail if the anomaly is sufficiently large. For $N = 100{,}000$, keeping an accepted deviation of 6.96% from the ideal value (see Def. 3.12), the acceptable range for the statistic $\mathscr{A}_k$ is $[46520, 53480]$. In our case, $\mathscr{A}_k$ is roughly equal to 32,000, so the Autocorrelation test from the AIS 20/31 fails quite significantly. On the other hand, our enhanced autocorrelation statistic is perfectly null for every lag $k > 0$, apart from the estimation errors.

We then produced a sequence affected solely by a correlation anomaly between bits distant of $k = 8$. More precisely, we have generated 100,000 bits of data affected by anomalies characterized by $\delta_8 = 0.6$ and $\xi = 0$. The analysis of this sequence then leads to the two graphs shown in Fig. 21.

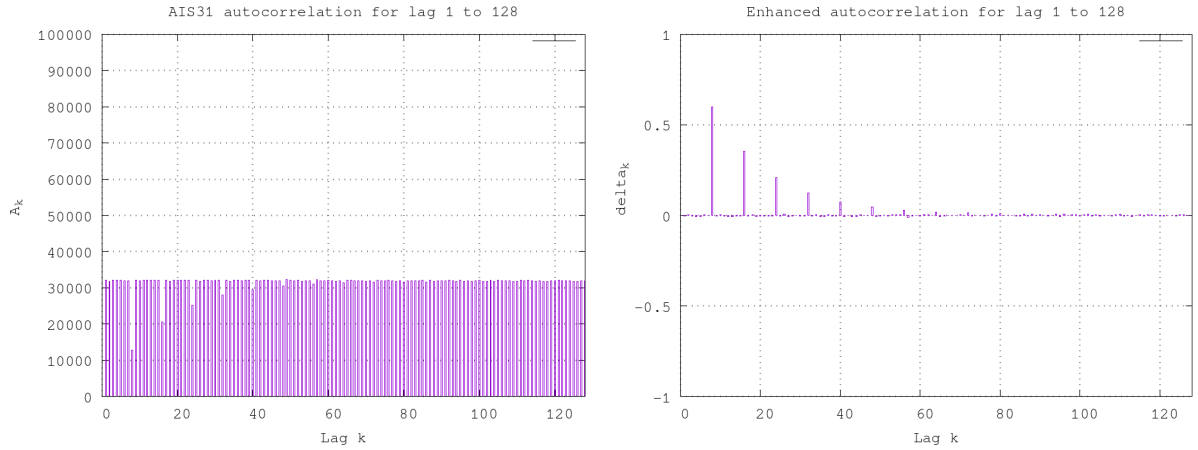
When the sequence is only affected by a correlation anomaly, both statistics give identical results, apart from the inversion of the "sign" of the anomaly, which is normal in view of the expected value of the autocorrelation statistic of the AIS 20/31 (see the proof of Th. 6.3). One of the remarkable points, however, is the fact that although the induced correlation anomaly acts on bits distant of $k = 8$, this anomaly propagates and detected between bits distant of every $k$ which is a multiple of 8, with an amplitude that appears to decrease exponentially with the lag. This phenomenon is intuitively understandable, but we will demonstrate it formally later.

e




(A) Autocorrelation statistic of the AIS 20/31 (B) Enhanced autocorrelation statistic

FIGURE 21. Autocorrelation statistics $\mathscr{A}_k$ vs $\mathscr{A}_k^\star$, $N = 100{,}000$, $\xi = 0$, $\delta_8 = 0.6$




(A) Autocorrelation statistic of the AIS 20/31 (B) Enhanced autocorrelation statistic

FIGURE 22. Autocorrelation statistics $\mathscr{A}_k$ vs $\mathscr{A}_k^\star$, $N = 100{,}000$, $\xi = 0.6$, $\delta_8 = 0.6$

Finally, we produced a sequence affected by the two anomalies (frequency and correlation), with characteristic parameters of $\xi = 0.6$ and $\delta_8 = 0.6$, which led to the graphs in Fig. 22.

As announced, when the two anomalies are present conjointly in the sequence, the autocorrelation statistic from the AIS 20/31 is well outside the acceptable range for any lag $k > 0$, and fails to correctly characterize the actual correlation anomaly for $k = 8$. In contrast, our enhanced autocorrelation statistic is not impacted by the global frequency anomaly, and still manages to perfectly characterize the correlation anomaly, judging by the almost identical appearance of graphs 21b and 22b.

**7.3.3. Sequences generated by a TRNG implemented on FPGA.** Finally, after validating our model on simulated sequences with known anomalies, and confirming the interest of our enhanced autocorrelation statistic compared to the statistic of the AIS 20/31, we wanted to study the use of the statistic in the case of real data issued from a physical generator. More precisely, we tested our statistic on an implementation of the Elementary RO-TRNG of Baudet *et al.* [**7**], developed for the OpenTRNG Project [**17**] and implemented on the Xilinx Artix-35T FPGA. The idea is to compare our results with those predicted by the model provided by Baudet *et al.*

As a reminder, the architecture of the Elementary RO-TRNG is described in Fig. 5, and is based on two ring oscillators, with an odd number of inverters $Inv_1$ and $Inv_2$ . In our case, we have set the number of inverters to $Inv_1 = 11$ and $Inv_2 = 7$. As explained in the introduction, the randomness produced by the Elementary RO-TRNG is based on clock jitter, which must accumulate over a certain period of time. The choice of prime numbers of inverters ensures that the jitter will have time to accumulate before the produced sequence repeats itself. We then made the frequency division factor $D$ vary to verify that increasing its value does indeed lead to an improvement in the quality of the randomness, which should translate into a reduction in the amplitude of statistical anomalies.

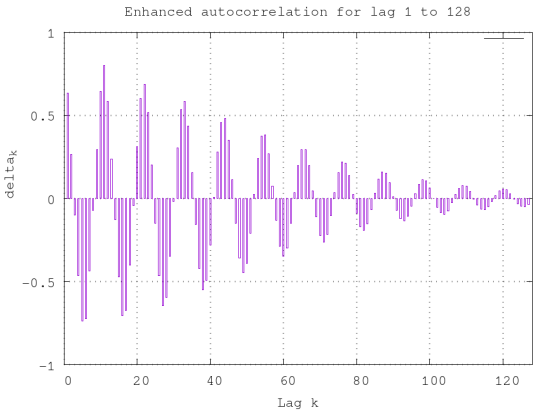
The Elementary RO-TRNG implementation we use produces blocks of $2^{14} = 16{,}384$ bits of data. We have therefore chosen to generate $7 \times 2^{14} = 114{,}688$ bits of data to be close to the quantity of data used in subsection 7.3.2 and to be able to compare the results we had obtained with those we obtain here.

> *Remark :* For an unknown reason, for any parameter $D$, and any parameter configuration $Inv_1$ and $Inv_2$ , the proportion of bits 1 was always equal to approximately 48.4%, which leads to $\xi = -0.032$. Although the consistency of this anomaly may raise questions, it appears in practice that its impact on the autocorrelation statistic of the AIS 20/31 (see the proof of Th. 6.3) as well as on the variance of the enhanced autocorrelation statistic (see section 7.2) is negligible.

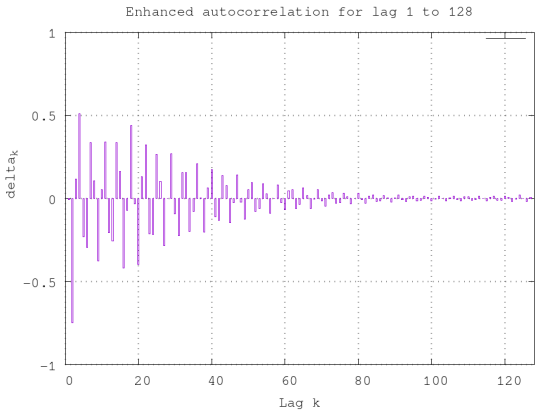
In view of the low amplitude of the frequency anomaly in the sequences produced, the results obtained with the statistic from the AIS 20/31 and with our enhanced statistic would be quite similar, as seen in the previous subsection. For this analysis, we have therefore only represented the results obtained with our enhanced autocorrelation statistic.

To analyze the Elementary RO-TRNG, we made the frequency division factor $D$ vary between 50 and 10,000. According to the model of Baudet *et al.* (recalled in section 3.1.2), increasing the factor $D$ should lead to a significant increase in entropy. Any anomaly that would be present for low values of $D$ should therefore disappear for larger values of this factor.

For low values of $D$, very strong correlations can be observed between successive bits, as shown in Fig. 23. The "shape" of the correlations varies as a function of $D$, and we have chosen to represent the graphs for which the correlations were the most noticeable. On the

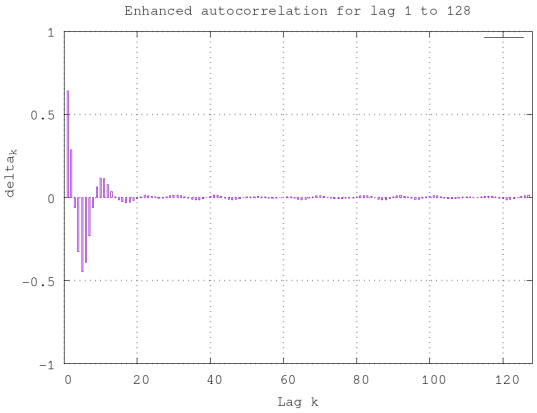



(A) $D = 50$

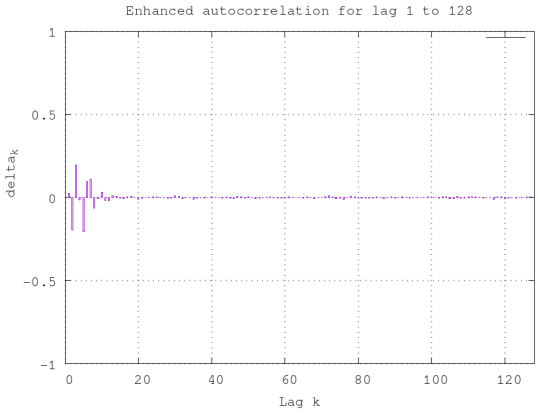



(B) $D = 84$

FIGURE 23. Enhanced autocorrelation statistic, Elementary RO-TRNG, $50 \leqslant D \leqslant 100$




(A) $D = 500$

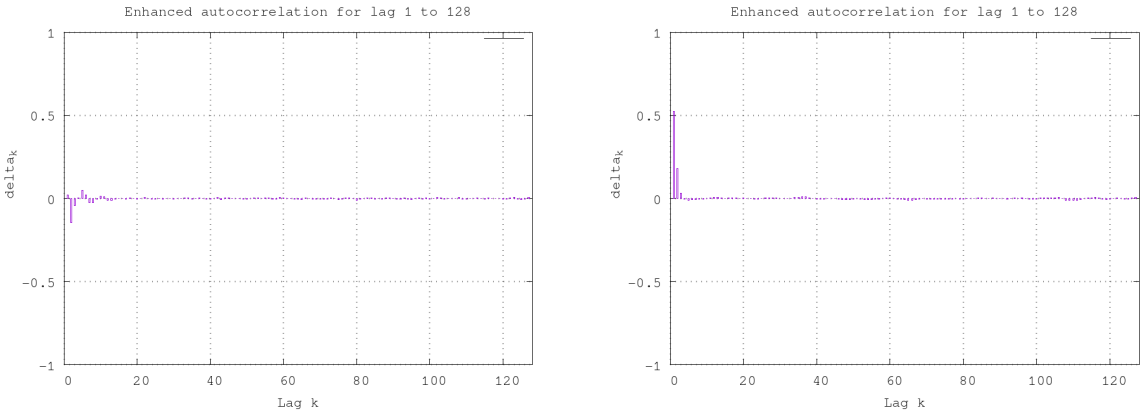



(B) $D = 535$

FIGURE 24. Enhanced autocorrelation statistic, Elementary RO-TRNG, $500 \leqslant D \leqslant 500$

other hand, what is already observable is that increasing the factor $D$ indeed leads to weaker correlations, both in their amplitude and in the distance up to which they are significant.

For larger values of $D$, here $500 \leqslant D \leqslant 550$, we find vaguely the same shapes of correlation graphs, depicted in Fig. 24, but these correlations decrease much more rapidly. However, it appears that for certain values of $D$ in this range, e.g. $D = 500$, the amplitude of the correlation anomaly is quite comparable to the one observed for $50 \leqslant D \leqslant 100$.




(A) $D = 2000$ (B) $D = 2034$

FIGURE 25. Enhanced autocorrelation statistic, Elementary RO-TRNG, $2000 \leqslant D \leqslant 2050$

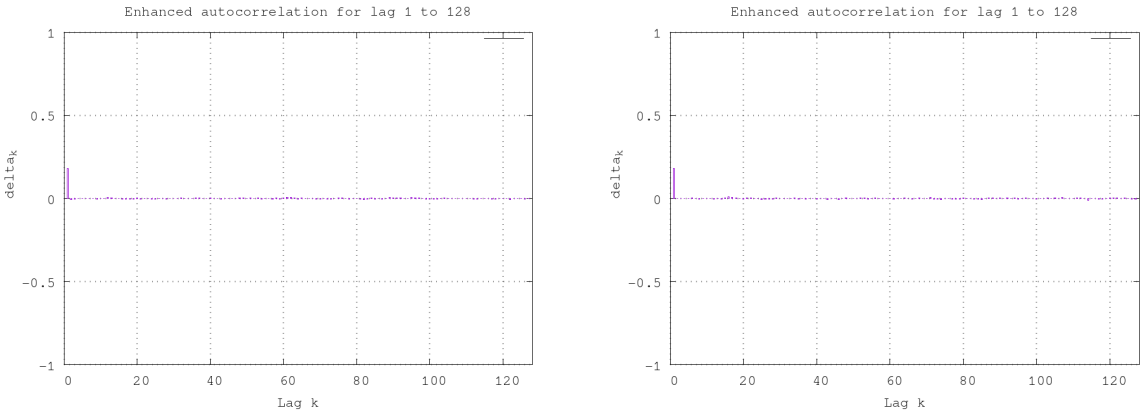
For even larger values of the factor $D$ (see Fig. 25), the autocorrelation phenomenon is practically non-existent, except for very small lags, typically $k = 1$ or 2. On the other hand, the amplitude of this anomaly again remains very high for certain values of the factor $D$, as can be seen in Fig. 25b.

To try to verify if this correlation anomaly ends up disappearing with the increase of $D$ as predicted by the model of Baudet *et al.* , we then pushed to the range $9950 \leqslant D \leqslant 10{,}000$. This led to the appearance of a single peak of correlation between bits distant of $k = 1$ and of amplitude $\delta_1 \approx 0.2$ for any frequency division factor $D$ in the range.

This discrepancy between the expected results and the observed graphs may be explained by two factors. A first possible explanation is that the implementation we used suffers from a correlation defect that was simply masked by the anomalies intrinsic to the architecture of the TRNG, and that this defect could only appear when the expected anomalies had disappeared. However, this hypothesis does not seem very credible, especially given that, for the interval $2000 \leqslant D \leqslant 2050$ in particular, several sequences showed very little autocorrelation (see graph 25a), and in particular no significant correlation for $k = 1$, like the one that can be seen in Fig. 26.

A second explanation is that there is a mismatch between the model introduced in [**7**] and the statistical properties of the data generated in practice, and that we have placed ourselves in a case that the model does not account for.

In our case, the discrepancy between the model and the data may stem from the fact that the model does not take into account correlated noises such as flicker noise. As explained in [**58**] and in the introduction of this manuscript, flicker noise is pink noise (also known as




(A) $D = 9950$ (B) $D = 9984$

FIGURE 26. Enhanced autocorrelation statistic, Elementary RO-TRNG, $9950 \leqslant D \leqslant 10{,}000$

$1/f$ noise) that prevails at low frequencies, while thermal noise (white, uncorrelated noise) prevails at high frequencies. This means that for longer accumulation times, and therefore for higher values of the factor $D$, the proportion of flicker noise to thermal noise increases. Hajimiri *et al.* then effectively demonstrated in [**35**] (Fig. 4) that clock jitter progressively becomes correlated as flicker noise takes precedence over thermal noise. The stochastic model of the ERO-TRNG having been developed for cases where the clock jitter was very small compared to the sampled signal period[1], this could explain why this correlation peak appears for large values of $D$.

The enhanced autocorrelation statistic therefore enabled us to visualize the limits of a theoretical model, by comparing it with the statistical properties of real data. In this way, we were able to demonstrate the interest of this statistic for evaluators who aim at testing the validity of a stochastic model, as well as for designers wishing to develop and verify a correlation model for their TRNG, or to find the limiting use cases of an existing model.

[1] [**7**], §2.3: "*For real physical systems, we expect the jitters to be small, that is,* $\sigma^2 \ll \mu$", $\sigma^2$ being the variance of the jitter, and $\mu$ the mean period of the signal.

CHAPTER 8

# Application of the enhanced autocorrelation to an attack on PTRNG

So far, we have demonstrated the optimality of our enhanced autocorrelation statistic for the characterization of bit-to-bit correlations in a binary sequence (section 7.1), then evaluated the effectiveness of this new statistic by applying it to binary sequences for which the characteristics of the anomalies were known, or from a generator whose parameters we had control over.

As we explained earlier, an adequate statistic can allow a designer to understand the impact of different parameterizations of his generator on the quality of the output sequences. For example, we have been able to analyze the impact of modifying the frequency division factor $D$ on the quality of the numbers produced, and, in particular, on their correlation. Another use case we will explore in this section, is the evaluation of the statistical characteristics of the sequences produced by the generator when it is led to work outside its nominal operating parameters. In particular, it may be interesting for an attacker to have a means to verify the success or failure of an attack by directly studying its impact on the sequence's statistical properties.

Considering that the most secure random number generators combine a physical entropy source with a cryptographic post-processing (see the `PTG.3` class of the AIS 20/31), the surface and diversity of attacks on PTRNGs are very large. One of the most notable attacks on TRNGs, and especially RO-based TRNGs is fault attack using extreme temperatures, ray injection, power glitches or underpowering for example [**75, 54**].

Another widespread attack on RO-based TRNGs is the *harmonic injection* or *frequency injection* attack. As a reminder, the quality of the randomness produced by a RO-based TRNG relies, at least, on the following two points: the amount of jitter must be sufficient, and the signals from the ROs (data signal and sampling signal) must be desynchronized, otherwise repeating patterns or, more generally, correlations between the data signal and the sampling signal could appear. The goal of a harmonic injection attack is then to force the signals to synchronize by injecting an additional signal of a given frequency into the generator, so as to cause a coupling effect [**36**] between this injected signal and the generator's useful signals. This additional signal can be injected through various means, the most common being the power supply [**52, 79, 16**], or by means of an electromagnetic probe [**52, 9, 8**]. If the injection is successful, the coupling effect will force the signals to operate at a certain frequency, linked to the injection frequency. In the context of an attack on the ERO-TRNG, the idea is to inject

a frequency that is an harmonic of both the data signal and the sampling signal, in order to synchronize both and provoke the appearance of correlations, or even pattern repetition, in the data produced by the generator.

> *Remark:* Often, the data and sampling oscillators operate at relatively close frequencies, and the harmonic injection attack will then simply attempt to make them oscillate at the same frequency.

In practice, when evaluating a real component, the frequency of the two oscillators of the ERO-TRNG is often unknown to the evaluator. The main challenge is thus to find a frequency that will enable this coupling effect with the oscillators. In [**8**], Bayon *et al.* performed a side channel analysis with their electromagnetic probe to extract the characteristics of the TRNG, in particular the frequency of its ring oscillators, before carrying out a fine-tuned injection attack based on this knowledge on the generator.

The attack we present in this chapter is an injection attack similar to the one presented by Bayon *et al.* but without the prior side channel analysis of the TRNG. The goal here is to find a frequency band for which the attack is a success based on the characteristics of the sequences produced by the generator, rather than based on the characteristics of the generator itself.

As mentioned earlier, in the event that the injection caused a coupling of the ring oscillators, that we call a *locking* of the oscillators, we expect the output data to be correlated. To judge the success of such an attack, we then need to be able to characterize the level of correlation of the data in order to detect a significant deviation from the case of uncorrelated data. One way of proceeding is to represent the data in the form of a binary matrix of pixels (often referred to as a *bitmap*), so that the values of the consecutive bits can be quickly observed (see Fig. 27).

Correlation between bits of the sequence will then make black or white "stripes" appear on the bitmap representation, which can become easily visible. On the other hand, this method has several drawbacks. Firstly, if the correlations introduced by the attack are not very significant, the phenomenon may be difficult to visualize on the bitmap. Secondly, if the attack introduces correlations of order greater than 1 (correlations between $B_{i-k}$ and $B_i$, for $k \geqslant 2$), the phenomenon will also be more difficult to detect. Finally, the major drawback of this method for real-life evaluation is that it is impossible to automate the process, since it is based on naked-eye observation of figures.

With this in mind, the enhanced autocorrelation statistic we have developed seems perfectly suited to achieve a more efficient detection of the success of the attack. To demonstrate this, we attempted to attack a real ERO-TRNG[1], by injecting an electromagnetic signal by means of a probe as described in Fig. 28 and illustrated in practice by the picture in Fig. 29.

[1] The data produced during this attack are property of the CEA.

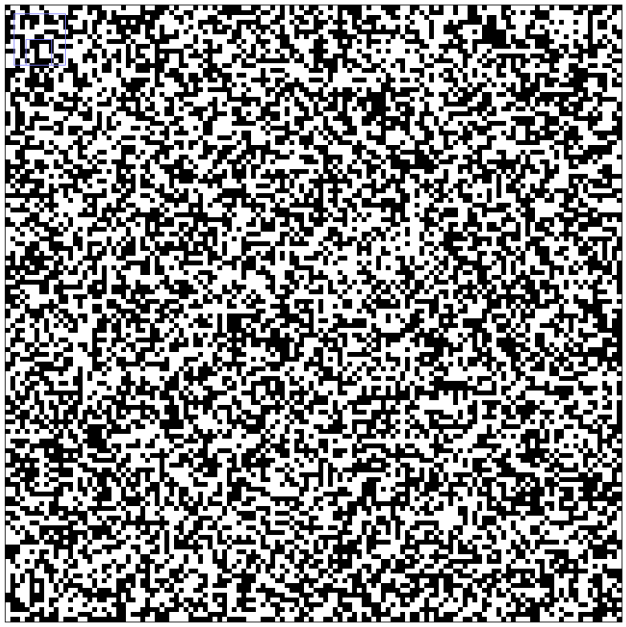


FIGURE 27. Bitmap representation of binary data

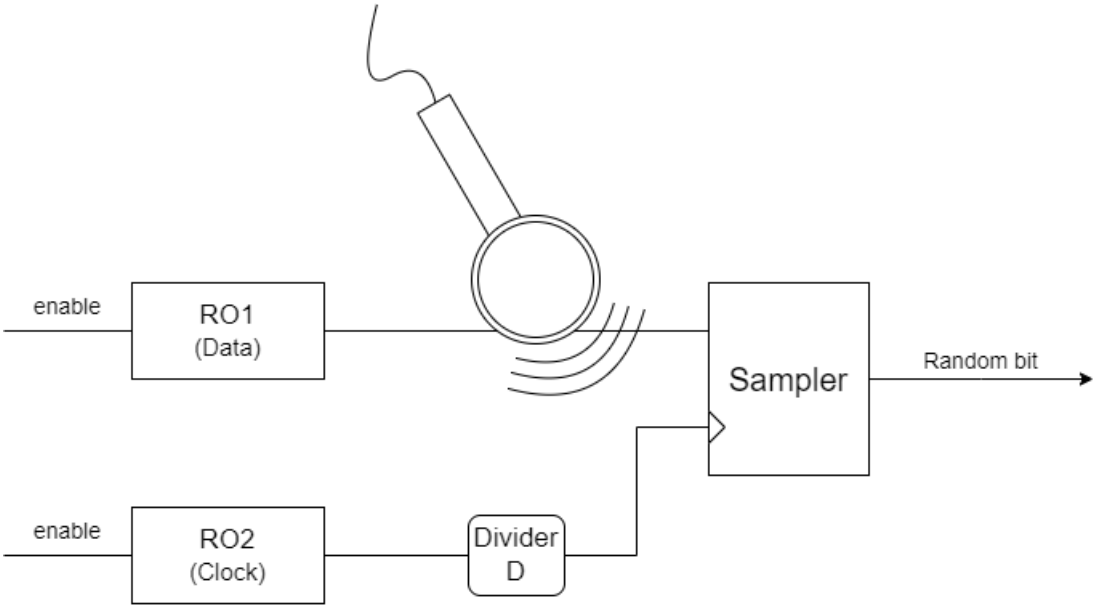



FIGURE 28. Schematic view of the harmonic injection attack on the ERO-TRNG

As the frequencies of the data and sampling oscillators were unknown, we made the frequency of the injected signal vary between 800 and 1000 MHz in steps of 0.1 MHz, in an attempt to

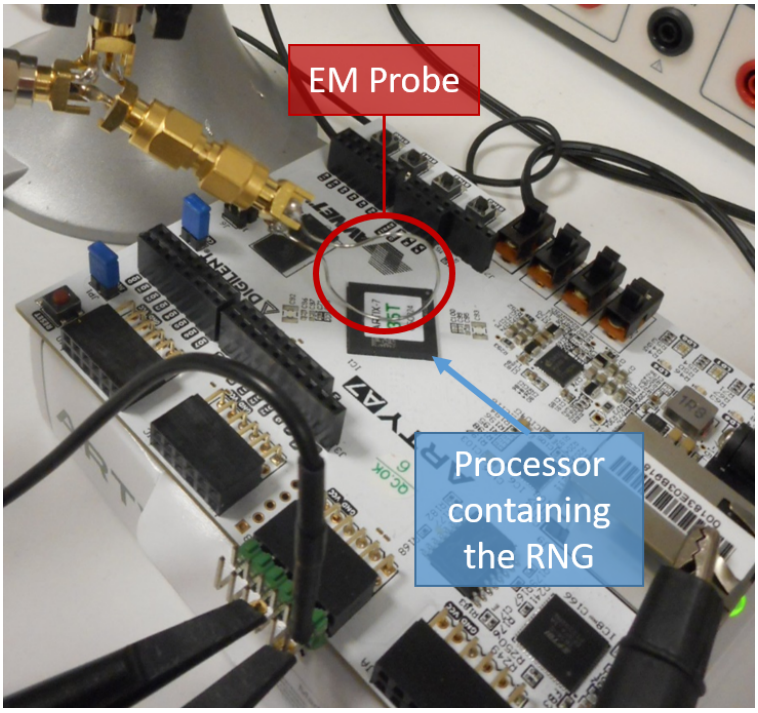



FIGURE 29. Bench of the harmonic injection attack on the ERO-TRNG

find the frequency band that would cause the two signals to synchronize. To automate the detection of the success of the attack, we first had to find the accepted autocorrelation threshold (the threshold for $\mathscr{A}_k^\star$ which is not exceeded in the absence of correlation). To minimize the variance of the estimation and thus limit the risk of a false detection of the locking, we chose to place a threshold on the mean value of the first 30 terms values of $\mathscr{A}_k^\star$, rather than on the individual values, with the hypothesis that the locking should cause correlation in bits relatively close to one another.

This threshold for the mean values can be obtained by computing the variance of the statistic $\mathscr{A}_k^\star$, or even through the computation of a *signal-to-noise ratio* [**72**], with a prior computation of the "noise" on data before the attack (expected to be completely uncorrelated). But we chose to place it empirically at a value of 0.003, considering that the generator produces 65,536 bits of data.

Once the threshold had been decided, we generated a set of data per frequency of the injected signal, then applied our enhanced autocorrelation statistic, which successfully provided us with a frequency band between 890 and 926 MHz. As anticipated, the locking attack caused significant correlations to appear between the generated bits. Figure 30 then represents the values $(\mathscr{A}_1^\star + ... + \mathscr{A}_{30}^\star)/30$ for different frequencies of the injected signal.

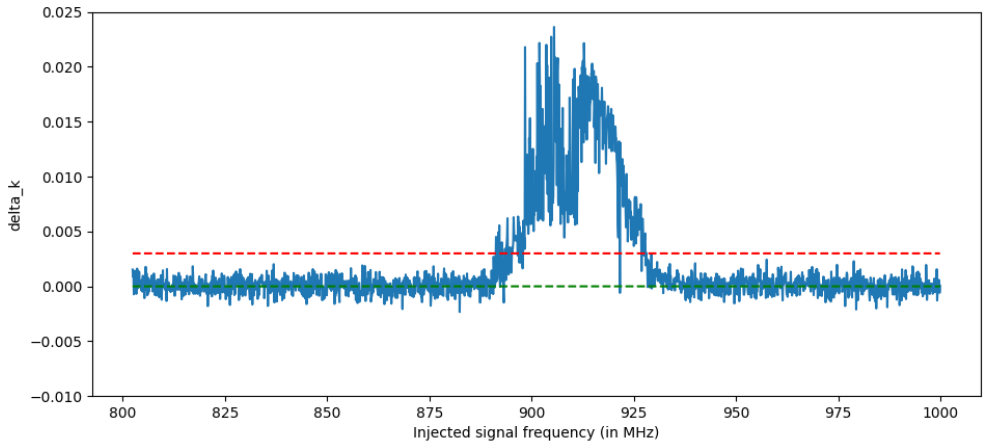
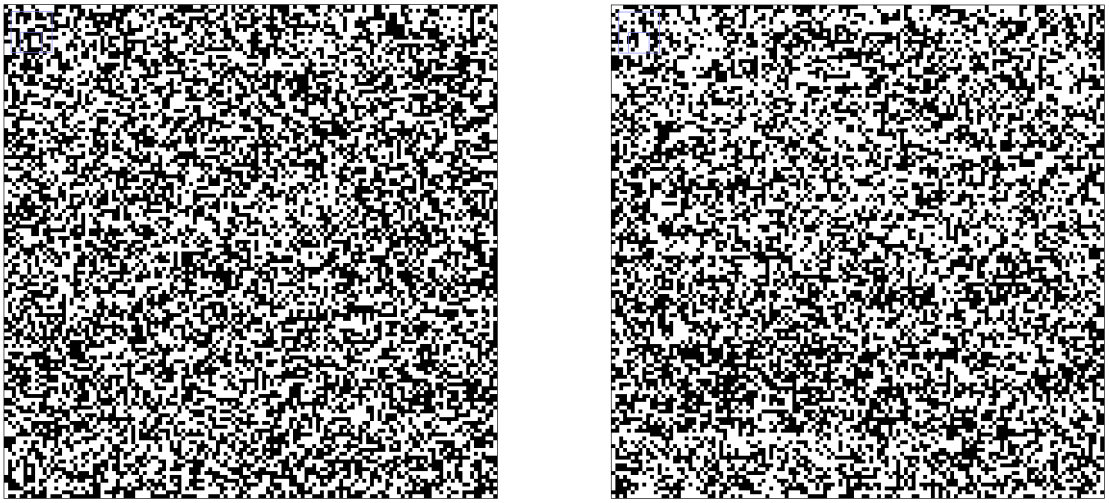



FIGURE 30. Mean of $\{\mathscr{A}_1^\star, ..., \mathscr{A}_{30}^\star\}$ at different injection frequencies

> *Remark:* In practice, the locking phenomenon occurs in a wider frequency band than the one detected by our threshold of correlation (up to around 936 MHz). But it can be argued that the detection with the threshold is accurate enough, with only a 22% difference between the frequency band obtained with the threshold, and the one for which the locking is successful.

The autocorrelation phenomenon resulting from the frequency injection is thus very easily visible in Fig 30, confirming the interest of the enhanced autocorrelation statistic in the context of such an attack. Previously, on the other hand, the detection of the success of this attack was carried out using the naked-eye visualization method described above. By observing the bitmaps derived from the generated data for both an injection frequency for which the attack fails and for a frequency for which it is successful (Fig. 31, zoomed on a part of the data), it appears that the correlations are barely visible, as the two bitmaps cannot be clearly distinguished. The presence of correlation is depicted by the appearance of horizontal stripes, most obvious on the lower part of the figure 31b.



(A) 875.0 MHz (B) 898.4 MHz

FIGURE 31. Bitmap representation of the data after frequency injection

It therefore seems all the more interesting to have a precise and numerical method to characterize correlations in an evaluation framework such as the enhanced autocorrelation statistic we propose here, in order to be able to decide on the success of an attack, even in the event that the impact of this attack on the statistical properties of the data is not extremely visible.

> *Remark:* In retrospect, although in our specific context, the mean of $\{A_1^\star, ..., A_{30}^\star\}$ allows for a clear visualization of the impact of the attack on the TRNG, it can be argued that other metrics can also be used, and could even be more relevant. In particular, the terms $A_k^\star$ can be positive as well as negative, and using their mean value could lead to values compensating each other and masking a potential correlation phenomenon (see Fig. 23 for an example of correlation "figures" which could lead to a low mean value for $\{A_1^\star, ..., A_{30}^\star\}$ despite each individual term being of high amplitude).
>
> To palliate this potential issue, for any new metric, we then recommend to use the **absolute values** $|A_k^\star|$ or the **squared values** $\left(A_k^\star\right)^2$ rather than the simple $A_k^\star$, so that every term is positive.
>
> Moreover, other functions of the statistics $|A_k^\star|$ (or $\left(A_k^\star\right)^2$) can be explored, such as the maximum value, which could lead to more glaring but also more variable results.
>
> Finally, the choice of the set of statistics $A_k^\star$ to use for our global metric can also be discussed. In our case, using the first 30 terms $A_k^\star$ proved to be sufficient to visualize the success of the attack. And although more terms could be included, it would lead to a global decrease of the mean value (as for higher $k$, the terms $A_k^\star$ are expected to be

of lower amplitude), and thus a much less visible distinction between a successful and a failed attack. Of course, with the use of another metric such as $\max_{1\leqslant k\leqslant n}|A_k^\star|$, lags much further than $k = 30$ can be considered without risk of "diluting" the information on correlations.

To visualize the appearance of correlations following a successful attack, we then propose the following (non-exhaustive) list of metrics:

- Mean of the $n$ first $|A_k^\star|$: $\frac{1}{n}\times\sum_{k=1}^{n}|A_k^\star|$, for $20\leqslant n\leqslant 30$,
- Mean of the $n$ first $\left(A_k^\star\right)^2$: $\frac{1}{n}\times\sum_{k=1}^{n}\left(A_k^\star\right)^2$, for $20\leqslant n\leqslant 30$,
- Median of the $n$ first $|A_k^\star|$: $\text{med}\left[\{A_1^\star,...,A_n^\star\}\right]$, for $20\leqslant n\leqslant 30$,
- Maximum of the $n$ first $|A_k^\star|$: $\max_{1\leqslant k\leqslant n}|A_k^\star|$.

However, this discussion on the metrics was performed after the experimentation, and the dataset produced during the attack was not accessible anymore. As such this list of metrics has not been evaluated, although their application to the data from the attack is expected to yield to similar results than those depicted in Fig. 30, in the form of a rough Bell curve, with significant correlations detected in the range [890MHz, 926MHz].

CHAPTER 9

# Towards a finer analysis of correlations

## Contents



In subsection 7.3.2, we saw that the presence of a phenomenon of correlation between bits distant of $k > 0$ invariably leads to the appearance of correlations between bits distant of all multiples of $k$.

Although, from the point of view of anomaly detection, these propagated correlation phenomena do not interfere with the result (as we will show later), from the point of view of precise characterization of the anomalies, the presence of this "artificial" information could interfere with the analysis of design defects of the generator that led to the anomaly (architecture, parameterization, ...).

In this section, we will propose ways of analyzing correlation anomalies in greater details, using new models and statistics.

### 9.1. Propagation of the correlation phenomena

Before proposing a new model, it is interesting to first understand where the propagated correlation phenomena come from. To do this, it seems natural to study the expression of the autocorrelation of order $mk$, $k > 0$, $m > 0$ as a function of the autocorrelation of order $k$.

As a reminder, the autocorrelation of order $k$ is the correlation between bits distant of $k$ and is expressed, for a stationary discrete stochastic process $\{X_i\}$, by:

$$\rho_k = \frac{\mathbb{E}\left[(X_i - \mu)(X_{i-k} - \mu)\right]}{\mathrm{Var}\,(X_i)},$$

and we have similarly the expression of the autocorrelation of order $mk$:

$$\rho_{mk} = \frac{\mathbb{E}\left[(X_i - \mu)(X_{i-mk} - \mu)\right]}{\mathrm{Var}\,(X_i)}.$$

To make a link between the theoretical autocorrelation of lag $mk$, $\rho_{mk}$, and the term $\delta_k$ from our correlation model, as we did in section 5.2, we use the following lemma:

LEMMA 9.1. *With the model considered for the correlation anomaly between bits distant of $k > 0$, for all $m \in \mathbb{N}$, $m > 0$:*

$$\Pr_{i,i-mk}(1 \mid 0) = \frac{(1+\xi)(1-\delta_k^m)}{2},$$

*and*

$$\Pr_{i,i-mk}(0 \mid 1) = \frac{(1-\xi)(1-\delta_k^m)}{2}.$$

PROOF. We will then prove this lemma by recurrence:

- For $m = 1$, the model for the anomaly provides by definition

  $$\Pr_{i,i-k}(1 \mid 0) = \frac{(1+\xi)(1-\delta_k)}{2},$$

  and

  $$\Pr_{i,i-k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_k)}{2}.$$

- We now assume that the result is established for $m \geqslant 1$,

  $$\Pr_{i,i-mk}(1 \mid 0) = \frac{(1+\xi)(1-\delta_k^m)}{2},$$

  and

  $$\Pr_{i,i-mk}(0 \mid 1) = \frac{(1-\xi)(1-\delta_k^m)}{2},$$

  and thus, by stationarity, that

  $$\Pr_{i-k,i-(m+1)k}(1 \mid 0) = \frac{(1+\xi)(1-\delta_k^m)}{2},$$

  and

  $$\Pr_{i-k,i-(m+1)k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_k^m)}{2}.$$

We then have,

$$\begin{aligned}
\Pr_{i,i-(m+1)k}(1 \mid 0) = \quad & \Pr_{i,i-k}(1 \mid 0) \times \Pr_{i-k,i-(m+1)k}(0 \mid 0) \\
& + \Pr_{i,i-k}(1 \mid 1) \times \Pr_{i-k,i-(m+1)k}(1 \mid 0),
\end{aligned}$$

$$\begin{aligned}
= \quad & \frac{(1+\xi)(1-\delta_k)}{2} \times \left(1 - \frac{(1+\xi)(1-\delta_k^m)}{2}\right) \\
& + \left(1 - \frac{(1-\xi)(1-\delta_k)}{2}\right) \times \frac{(1+\xi)(1-\delta_k^m)}{2},
\end{aligned}$$

$$\begin{aligned}
= \quad & \frac{1+\xi}{2} \times \left[(1-\delta_k) \times \frac{1-\xi+\delta_k^m+\xi\delta_k^m}{2}\right. \\
& \left. + \frac{1+\xi+\delta_k-\xi\delta_k}{2} \times (1-\delta_k^m)\right],
\end{aligned}$$

$$\begin{aligned}
= \quad & \frac{1+\xi}{2} \times \left[\frac{1-\xi+\delta_k^m+\xi\delta_k^m-\delta_k+\xi\delta_k-\delta_k^{m+1}-\xi\delta_k^{m+1}}{2}\right. \\
& \left. + \frac{1+\xi+\delta_k-\xi\delta_k-\delta_k^m-\xi\delta_k^m-\delta_k^{m+1}+\xi\delta_k^{m+1}}{2}\right],
\end{aligned}$$

$$= \quad \frac{(1+\xi)(1-\delta_k^{m+1})}{2}.$$

And we prove in a strictly analogous fashion that:

$$\Pr_{i,i-(m+1)k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_k^{m+1})}{2}.$$

We therefore proved the lemma by recurrence. □

THEOREM 9.2 (Autocorrelation of order $mk$ of the process $\{B_i\}$). *Let $\{B_i\}$ be the stationary discrete stochastic process describing the generation of bits according to the model defined in 5.2. Let $(k,m) \in \mathbb{N}^2$, $k > 0$, $m > 0$. Then the self-correlation of order $mk$ of $\{B_i\}$, denoted by $\rho_{mk}$, is equal to:*

$$\rho_{mk} = \delta_k^m.$$

PROOF. Similarly to the proof of Th. 5.3, we have:

$$\mathrm{Var}\,(B_i) = \mathrm{Pr}_i\,(0) \times \mathrm{Pr}_i\,(1) = \frac{1-\xi^2}{4},$$

and

$$\begin{aligned}\mathbb{E}\left[(B_i-\mu)(B_{i-mk}-\mu)\right] = {} & (1-\mu)(1-\mu) \times \mathrm{Pr}_{i,i-mk}\,(1 \mid 1)\,\mathrm{Pr}_{i-mk}\,(1) \\ & + (0-\mu)(1-\mu) \times \mathrm{Pr}_{i,i-mk}\,(0 \mid 1)\,\mathrm{Pr}_{i-mk}\,(1) \\ & + (1-\mu)(0-\mu) \times \mathrm{Pr}_{i,i-mk}\,(1 \mid 0)\,\mathrm{Pr}_{i-mk}\,(0) \\ & + (0-\mu)(0-\mu) \times \mathrm{Pr}_{i,i-mk}\,(0 \mid 0)\,\mathrm{Pr}_{i-mk}\,(0)\,.\end{aligned}$$

With the hypothesis of stationarity, we can replace $\mathrm{Pr}_{i-mk}\,(x)$, $x \in \{0,1\}$ by $\mathrm{Pr}_i\,(x)$, and $\mu$ by $\mathrm{Pr}_i\,(1)$ to obtain:

$$\begin{aligned}\mathbb{E}\left[(B_i-\mu)(B_{i-mk}-\mu)\right] = {} & \mathrm{Pr}_i\,(1)\,\mathrm{Pr}_i\,(0) \times \Big[\,\mathrm{Pr}_i\,(0)\,\mathrm{Pr}_{i,i-mk}\,(1 \mid 1) - \mathrm{Pr}_i\,(0)\,\mathrm{Pr}_{i,i-mk}\,(1 \mid 0) \\ & + \mathrm{Pr}_i\,(1)\,\mathrm{Pr}_{i,i-mk}\,(0 \mid 0) - \mathrm{Pr}_i\,(1)\,\mathrm{Pr}_{i,i-mk}\,(0 \mid 1)\,\Big],\end{aligned}$$

By simplifying the term $\mathrm{Pr}_i\,(0) \times \mathrm{Pr}_i\,(1)$, we get:

$$\begin{aligned}\rho_{mk} = {} & \mathrm{Pr}_i\,(0)\left[\,\mathrm{Pr}_{i,i-mk}\,(1 \mid 1) - \mathrm{Pr}_{i,i-mk}\,(1 \mid 0)\,\right] \\ & + \mathrm{Pr}_i\,(1)\left[\,\mathrm{Pr}_{i,i-mk}\,(0 \mid 0) - \mathrm{Pr}_{i,i-mk}\,(0 \mid 1)\,\right], \\ = {} & \big(\mathrm{Pr}_i\,(0) + \mathrm{Pr}_i\,(1)\big)\big[1 - \mathrm{Pr}_{i,i-mk}\,(0 \mid 1) - \mathrm{Pr}_{i,i-mk}\,(1 \mid 0)\,\big], \\ = {} & 1 - \big(\mathrm{Pr}_{i,i-mk}\,(0 \mid 1) + \mathrm{Pr}_{i,i-mk}\,(1 \mid 0)\big).\end{aligned}$$

Then, using lemma 9.1, we can then replace $\mathrm{Pr}_{i,i-mk}\,(0 \mid 1)$ and $\mathrm{Pr}_{i,i-mk}\,(1 \mid 0)$ by their expression according to our model, to finally prove the theorem:

$$\rho_{mk} = 1 - \left(\frac{(1-\xi)(1-\delta_k^m)}{2} + \frac{(1+\xi)(1-\delta_k^m)}{2}\right) = \delta_k^m.$$

□

The result of this theorem therefore implies that an autocorrelation phenomenon is always accompanied by "harmonic" phenomena, and that the amplitude of these harmonic correlations decreases exponentially (it decreases because $\delta_k \in ]-1,1[$), which is perfectly visible in Fig. 22b for example.

> *Remark :* This result also implies that a test which did not fail due to the presence of the original correlation phenomenon cannot fail due to the harmonic phenomena, as these are always of lower amplitude. The goal of concealing harmonic correlation phenomena

is therefore not to prevent unwarranted failures of a test, but simply to enable a designer to clearly characterize the real correlation phenomena from which his generator suffers.

## 9.2. Extending the model for correlations

The presence of "harmonic" correlation phenomena on Fig. 21 and 22 is due to the fact that the correlation model we have proposed only studies correlations between two bits at a time, as does the autocorrelation statistic of the AIS 20/31. In order to solve this problem, our objective is to develop at a global model, which will take into account information from $n \in \mathbb{N}$ preceding bits rather than from a single bit.

**9.2.1. Correlation model with the knowledge of two preceding bits.** To gain insight into a general model, we started by developing a correlation model that takes into account the information of two preceding bits. More specifically, for $k > 0$, we sought to develop a model for the following probabilities:

$$\Pr_{i,\ i-k,\ i-2k}\left(x \mid y, z\right),$$

where $(x, y, z) \in \{0,1\}^3$.

To do this, similarly to the methodology used to obtain the model established in Def. 5.2, we started from the general expression of a trivariate polynomial of degree 3 and tried to identify the set of coefficients of the polynomial.

To distinguish the parameters of our first "simple" model (Cf. Def. 5.2) from those of this more complex one, we use the notations $\delta_{2,k}$ and $\delta_{2,2k}$ for the terms describing the correlation between bits distant of $k$ and $2k$ respectively. The parameter $\xi$, on the other hand, remains unchanged, regardless of the choice of model, as it is independent of the correlation phenomena.

9.2.1.1. *Model for* $\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1)$*:* In all generality, the expression of $\Pr_{i,i-k,i-2k}(0 \mid 1,1)$ as a trivariate polynomial of degree 3 is:

$$\begin{aligned}\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1) = \quad & a_1\xi^3 + a_2\delta_{2,k}^3 + a_3\delta_{2,2k}^3 + b_1\xi^2\delta_{2,k} + b_2\xi\delta_{2,k}^2 + b_3\xi^2\delta_{2,2k} + b_4\xi\delta_{2,k}^2 \\ & + b_5\delta_{2,k}^2\delta_{2,2k} + b_6\delta_{2,k}\delta_{2,2k}^2 + c\xi\delta_{2,k}\delta_{2,2k} + d_1\xi^2 + d_2\delta_{2,k}^2 + d_3\delta_{2,2k}^2 \\ & + e_1\xi\delta_{2,k} + e_2\xi\delta_{2,2k} + e_3\delta_{2,k}\delta_{2,2k} + f_1\xi + f_2\delta_{2,k} + f_3\delta_{2,2k} + g,\end{aligned}$$

and we seek to find the value of the real coefficients $a_1, a_2, a_3, b_1, b_2, b_3, b_4, b_5, b_6, c,\ d_1, d_2, d_3$, $e_1, e_2, e_3, f_1, f_2, f_3$ and $g$.

First of all, we want the case where $\delta_{2,2k} = 0$ to characterize the absence of real correlation between bits distant of $2k$, and therefore the presence of a single potential real correlation between bits distant of $k$. In this sense, we want to fall back to the model established in Def. 5.2, i.e.:

$$\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1) = \Pr_{i,\ i-k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_{2,k})}{2}.$$

By identifying the polynomials, we then get $a_1 = a_2 = b_1 = b_2 = d_1 = d_2 = 0$, $f_1 = f_2 = -\frac{1}{2}$, and $e_1 = g = \frac{1}{2}$.

Similarly, when $\delta_{2,k} = 0$, we want to fall back to the model of a single potential correlation between bits distant of $k$, i.e.:

$$\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1) = \Pr_{i,\ i-2k}(0 \mid 1) = \frac{(1-\xi)(1-\delta_{2,2k})}{2},$$

which then provides $a_3 = b_3 = b_4 = d_3 = 0$, $f_3 = -\frac{1}{2}$ and $e_2 = \frac{1}{2}$.

For the time being, the expression can thus be simplified as follows:

$$\begin{aligned}\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1) = &\frac{(1-\xi)(1-\delta_{2,k}-\delta_{2,2k})}{2}\\ &+ b_5\delta_{2,k}^2\delta_{2,2k} + b_6\delta_{2,k}\delta_{2,2k}^2 + c\xi\delta_{2,k}\delta_{2,2k} + e_3\delta_{2,k}\delta_{2,2k}.\end{aligned}$$

To find the last coefficients, we make use of the fact that $\delta_{2,k} \to 1$ characterizes the case where the value of the bit of index $i-k$ is almost surely (in the probabilistic sense) equal to the value of the bit of index $i$, for any $i \in \mathbb{N}$. In other words, for any value of $\delta_{2,2k} \in ]-1,1[$:

$$\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1) \underset{\delta_{2,k}\to 1}{\to} 0.$$

Therefore, a reasoning at the limits (polynomials being continuous functions) provides:

$$-\frac{1-\xi}{2}\delta_{2,2k} + b_5\delta_{2,2k} + b_6\delta_{2,2k}^2 + c\xi\delta_{2,2k} + e_3\delta_{2,2k} = 0,$$

which leads to $c = -\frac{1}{2}$, $b_6 = 0$ and $b_5 + e_3 = \frac{1}{2}$.

Using the same reasoning for the case where $\delta_{2,2k} \to 1$ with $\delta_{2,k} \in ]-1,1[$, we get $b_5 = 0$ and $b_6 + e_3 = \frac{1}{2}$.

Combining the different results, we find in the end that $b_5 = b_6 = 0$ and $e_3 = \frac{1}{2}$. We have then identified all the coefficients of the generic polynomial.

Thus, the polynomial model that satisfies all the constraints we have set is the following:

$$\begin{aligned}\Pr_{i,\ i-k,\ i-2k}(0 \mid 1,1) &= \frac{(1-\xi)(1-\delta_{2,k})(1-\delta_{2,2k})}{2},\\ &= \Pr_{i,\ i-k}(0 \mid 1) \times (1-\delta_{2,2k}).\end{aligned}$$

9.2.1.2. *Model for* $\Pr_{i,\ i-k,\ i-2k}(1 \mid 0,0)$*:* For $\Pr_{i,\ i-k,\ i-2k}(1 \mid 0,0)$, a strictly identical reasoning leads to:

$$\begin{aligned}\Pr_{i,\ i-k,\ i-2k}(1 \mid 0,0) &= \frac{(1+\xi)(1-\delta_{2,k})(1-\delta_{2,2k})}{2},\\ &= \Pr_{i,\ i-k}(1 \mid 0) \times (1-\delta_{2,2k}).\end{aligned}$$

9.2.1.3. *Model for* $\Pr_{i,\ i-k,\ i-2k}(0 \mid 0,1)$*:* To try to have an insight on the generic case, it seems of course interesting to consider the case where the bits of index $i-k$ and $i-2k$ are different, for example when the bit of index $i-k$ is 0 and the bit of index $i-2k$ is 1.

Rather than trying to find the expression of the probabilities like we did in the previous two cases, it is wiser to rather use the following property, based on the law of total probabilities, which can be seen as an extension of Prop. 5.1:

PROPERTY 9.3. *Let* $\{B_i\}$ *be a binary-valued stationary stochastic process,* $(k,n) \in \mathbb{N}^2$*,* $k > 0$*,* $n > 1$*. We have:*

$$\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(0,1,...,1,1) = \Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,1,...,1,0)\,,$$

*and*

$$\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,0,...,0,0) = \Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(0,0,...,0,1)\,.$$

PROOF. In all generality:

$$\begin{aligned}&\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,1,...,1,1) + \Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,1,...,1,0)\\ &= \Pr_{i,\ i-k,\ ...,\ i-(n-1)k}(1,1,...,1)\,,\end{aligned}$$

or

$$\begin{aligned}&\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1 \mid 1,...,1,1) \times \Pr_{i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,...,1,1)\\ &+ \Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,1,...,1,0) = \Pr_{i,\ i-k,\ ...,\ i-(n-1)k}(1,1,...,1)\,.\end{aligned}$$

But, by stationarity:

$$\Pr_{i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,...,1,1) = \Pr_{i,\ i-k,\ ...,\ i-(n-1)k}(1,1,...,1)\,,$$

and this provides:

$$\begin{aligned}&(1-\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1\mid 1,...,1,1))\times\Pr_{i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,...,1,1)\\&=\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,1,...,1,0)\,,\end{aligned}$$

which, as announced, further simplifies to:

$$\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(0,1,...,1,1)=\Pr_{i,\ i-k,\ ...,\ i-(n-1)k,\ i-nk}(1,1,...,1,0)\,.$$

The bits 0 and 1 have perfectly symmetrical roles, so the proof for the second equation is identical and we leave it to the reader's discretion.

□

In our case, $n=2$. We thus have:

$$\Pr_{i,\ i-k,\ i-2k}(1,0,0)=\Pr_{i,\ i-k,\ i-2k}(0,0,1)\,,$$

which can be developed into:

$$\begin{aligned}(1-\Pr_{i,\ i-k,\ i-2k}(0\mid 0,0))\times\Pr_{i,\ i-k}(0\mid 0)\times\Pr_i(0)&\\=\Pr_{i,\ i-k,\ i-2k}(0\mid 0,1)\times\Pr_{i,\ i-k}(0\mid 1)\times\Pr_i(1)&\end{aligned}$$

With the models for correlation we have obtained up to now, the equation above translates into:

$$\begin{aligned}\frac{(1+\xi)(1-\delta_{2,k})(1-\delta_{2,2k})}{2}\times\Pr_{i,\ i-k}(0\mid 0)\times\frac{1-\xi}{2}&\\=\Pr_{i,\ i-k,\ i-2k}(0\mid 0,1)\times\frac{(1-\xi)(1-\delta_{2,k})}{2}\times\frac{1+\xi}{2}&\end{aligned}$$

Then, by identifying the coefficients, we directly obtain the expression of $\Pr_{i,i-k,i-2k}(0\mid 0,1)$, which is:

$$\begin{aligned}\Pr_{i,\ i-k,\ i-2k}(0\mid 0,1)&=\Pr_{i,\ i-k}(0\mid 0)\times(1-\delta_{2,2k}),\\&=\left(1-\frac{(1+\xi)(1-\delta_{2,k})}{2}\right)\times(1-\delta_{2,2k}).\end{aligned}$$

9.2.1.4. *Model for* $\Pr_{i,\ i-k,\ i-2k}(1 \mid 1,0)$ *:* Again, using the same reasoning as for the previous probability, we obtain the following expression for $\Pr_{i,\ i-k,\ i-2k}(1 \mid 1,0)$:

$$\Pr_{i,\ i-k,\ i-2k}(1 \mid 1,0) = \Pr_{i,\ i-k}(1 \mid 1) \times (1-\delta_{2,2k}),$$

$$= \left(1 - \frac{(1-\xi)(1-\delta_{2,k})}{2}\right) \times (1-\delta_{2,2k}).$$

Thus, the complete model is defined as such:

DEFINITION 9.4 (Correlation model with the knowledge of two preceding bits). *For $i \in \mathbb{N}$, $k > 0$, we model the correlation between the bit of index $i$ and the bits of indexes $i-k$ and $i-2k$ by the following set of equations:*

$$\Pr_{i,i-k,i-2k}(0 \mid 1,1) = \Pr_{i,i-k}(0 \mid 1) \times (1-\delta_{2,2k}),$$

$$\Pr_{i,i-k,i-2k}(1 \mid 0,0) = \Pr_{i,i-k}(1 \mid 0) \times (1-\delta_{2,2k}),$$

$$\Pr_{i,i-k,i-2k}(0 \mid 0,1) = \Pr_{i,i-k}(0 \mid 0) \times (1-\delta_{2,2k}),$$

$$\Pr_{i,i-k,i-2k}(1 \mid 1,0) = \Pr_{i,i-k}(1 \mid 1) \times (1-\delta_{2,2k}),$$

*which can be summarized by:*

$$\Pr_{i,i-k,i-2k}\left(x \mid y,\overline{x}\right) = \Pr_{i,i-k,i-2k}\left(x \mid y\right) \times (1-\delta_{2,2k}),$$

*where $(x,y) \in \{0,1\}^2$ and $\overline{x}$ is the complementary value of the bit $x$.*

> *Remark: Probabilities of the kind $\Pr_{i,i-k,i-2k}\left(x \mid y,x\right)$ are also described by this model, as we simply have:*
>
> $$\Pr_{i,i-k,i-2k}\left(x \mid y,x\right) = 1 - \Pr_{i,i-k,i-2k}\left(\overline{x} \mid y,x\right).$$

**9.2.2. Proposal for a global model for correlations.** From this definition, and especially in view of the calculations that led to it, a conjecture about the general form for a correlation model in the knowledge of $n > 0$ preceding bits seems to emerge. For the sake of simplicity, and since we are talking about a global model here, we will set $k = 1$ without loss of generality.

> **Notation:** To not impair readability we will use the following notation for conditional probabilities with the knowledge of $n$ preceding bits:
>
> $$\Pr\left(B_i = b_i \mid B_{i-1} = b_{i-1}, ..., B_{i-n} = b_{i-n}\right) = \Pr_{i \to i-n}\left(b_i \mid b_{i-1}, ..., b_{i-n}\right).$$

DEFINITION 9.5 (Correlation model with the knowledge of $n$ preceding bits). *For $i \in \mathbb{N}$, $n > 0$, we propose to model the correlation between the bit of index $i$ and the bits of indexes $i-1$, $i-2$, ..., $i-n$ by the following equation:*

$$\Pr_{i \to i-n}\left(b_i \mid w_{n-1}, \overline{b_i}\right) = \Pr_{i \to i-n+1}(b_i \mid w_{n-1}) \times (1 - \delta_{n,n})$$

*where $b_i \in \{0,1\}$, $w_{n-1} = (b_{i-1}, ..., b_{i-n+1}) \in \{0,1\}^{n-1}$ and $\delta_{n,n} \in ]-1,1[$.*

The global model with the knowledge of $n > 0$ bits is then a set of $2^n$ equations, and can be seen as a natural extension of the model with the knowledge of $n-1$ preceding bits.

**9.2.3. Constraints on the parameters of the model.** In the same fashion that the parameter $\delta_k$ of the "simple" correlation model established in Def. 5.2 is constrained by the value of $\xi$, it quickly appears that the parameter $\delta_{n,n}$ of our new model is also constrained in the values it can take. More specifically, we have the following property:

PROPERTY 9.6 (Constraints on the value of $\delta_{n,n}$). *For all $b_i \in \{0,1\}$ and $w_{n-1} = (b_{i-1}, ..., b_{i-n+1}) \in \{0,1\}^{n-1}$, we have the following inequation:*

$$-\frac{\Pr_{i \to i-n+1}\left(\overline{b_i} \mid w_{n-1}\right)}{\Pr_{i \to i-n+1}(b_i \mid w_{n-1})} \leqslant \delta_{n,n} \leqslant 1.$$

PROOF. Let $b_i \in \{0,1\}$ and $w_{n-1} = (b_{i-1}, ..., b_{i-n+1}) \in \{0,1\}^{n-1}$. $\Pr_{i \to i-n}\left(b_i \mid w_{n-1}, \overline{b_i}\right)$ being a probability, we have, in all generality:

$$0 \leqslant \Pr_{i \to i-n}\left(b_i \mid w_{n-1}, \overline{b_i}\right) = \Pr_{i \to i-n+1}(b_i \mid w_{n-1}) \times (1 - \delta_{n,n}) \leqslant 1.$$

This then translates into:

$$1 - \frac{1}{\Pr_{i \to i-n+1}(b_i \mid w_{n-1})} \leqslant \delta_{n,n} \leqslant 1,$$

or, as announced:

$$-\frac{\Pr_{i \to i-n+1}\left(\overline{b_i} \mid w_{n-1}\right)}{\Pr_{i \to i-n+1}(b_i \mid w_{n-1})} \leqslant \delta_{n,n} \leqslant 1.$$

because $1 - \Pr_{i \to i-n+1}(b_i \mid w_{n-1}) = \Pr_{i \to i-n+1}\left(\overline{b_i} \mid w_{n-1}\right)$.

□

**9.2.4. Finer autocorrelation statistic.** Although the model we propose does not directly provide an explicit expression for the conditional probabilities (even though this expression can be obtained by developing step by step), it does allow a very simple estimation of the relevant parameter $\delta_{n,n}$. Indeed, very similarly to the model for a single correlation phenomenon between bits distant of $k > 0$, it is very easy to extract $\delta_{n,n}$ from the conditional probabilities of the model. More precisely, we have the following property:

PROPERTY 9.7. *For all $w_{n-1} \in \{0,1\}^{n-1}$, we have:*

$$\Pr_{i \to i-n}(0 \mid w_{n-1}, 1) + \Pr_{i \to i-n}(1 \mid w_{n-1}, 0) = 1 - \delta_{n,n}.$$

PROOF. Let $n > 1$ and $w_{n-1} \in \{0,1\}^{n-1}$. By definition of the model:

$$\begin{aligned} \Pr_{i \to i-n}(0 \mid w_{n-1}, 1) + \Pr_{i \to i-n}(1 \mid w_{n-1}, 0) = & \quad [\Pr_{i \to i-n}(0 \mid w_{n-1}) + \Pr_{i \to i-n}(1 \mid w_{n-1})] \\ & \times (1 - \delta_{n,n}), \\ = & \quad (1 - \delta_{n,n}). \end{aligned}$$

□

It is then very easy to develop an unbiased estimator of $\delta_{n,n}$, and thus allows for the definition of an optimal test for the global correlation anomaly as described by our new model.

DEFINITION 9.8 (Finer autocorrelation statistic given $w_{n-1}$). *For $n > 1$ and $w_{n-1} = b_{i-n+1}, ..., b_{i-1} \in \{0,1\}^{n-1}$ chosen such that at least one occurrence of $w_{n-1}$ is preset by a bit 0, and another by a bit 1, we define the **finer autocorrelation statistic given** the observation of the $(n-1)$-bit word $\boldsymbol{w_{n-1}}$ as follows:*

$$\mathscr{F}_n^\star \mid w_{n-1} = 1 - \frac{N_{1,w_{n-1},0}}{N_{1,w_{n-1}}} - \frac{N_{0,w_{n-1},1}}{N_{0,w_{n-1}}},$$

*where $(x,y) \in \{0,1\}^2$ and $N_{x,w_{n-1}}$ and $N_{x,w_{n-1},y}$ represent the number of occurrences of the words $(x, b_{i-n+1}, ..., b_{i-1})$ and $(x, b_{i-n+1}, ..., b_{i-1}, y)$ respectively.*

We then have the following theorem:

THEOREM 9.9 (Expected value of $\mathscr{F}_n^\star \mid w_{n-1}$). *For any $n > 1$ and $w_{n-1} = b_{i-n+1}, ..., b_{i-1} \in \{0,1\}^{n-1}$. We have:*

$$\mathbb{E}\left[\mathscr{F}_n^\star \mid w_{n-1}\right] = \delta_{n,n}.$$

*In other words, $\mathscr{F}_n^\star \mid w_{n-1}$ is an unbiased estimator of $\delta_{n,n}$.*

PROOF. Let $n > 1$ and $w_{n-1} = b_{i-n+1}, ..., b_{i-1} \in \{0,1\}^{n-1}$. In all generality:

$$\mathbb{E}\left[\mathscr{F}_n^\star \mid w_{n-1}\right] = 1 - \mathbb{E}\left[\frac{N_{1,w_{n-1},0}}{N_{1,w_{n-1}}}\right] - \mathbb{E}\left[\frac{N_{0,w_{n-1},1}}{N_{0,w_{n-1}}}\right].$$

Focusing on the first expectation, we have:

$$\begin{aligned}
\mathbb{E}\left[\frac{N_{1,w_{n-1},0}}{N_{1,w_{n-1}}}\right] &= \sum_{n_1=1}^{n_{w_{n-1}}-1} \sum_{n_{10}=0}^{n_1} \frac{n_{10}}{n_1} \times \Pr\left(N_{1,w_{n-1},0} = n_{10}, N_{1,w_{n-1}} = n_1\right), \\
&= \sum_{n_1=1}^{n_{w_{n-1}}-1} \Pr\left(N_{1,w_{n-1}} = n_1\right) \sum_{n_{10}=1}^{n_1} \frac{n_{10}}{n_1} \times \Pr\left(N_{1,w_{n-1},0} = n_{10} \mid N_{1,w_{n-1}} = n_1\right), \\
&= \sum_{n_1=1}^{n_{w_{n-1}}-1} \Pr\left(N_{1,w_{n-1}} = n_1\right) \sum_{n_{10}=1}^{n_1} \binom{n_1-1}{n_{10}-1} \\
&\quad \times \Pr_{i\to i-n}(0 \mid w_{n-1}, 1)^{n_{10}} \left(1 - \Pr_{i\to i-n}(0 \mid w_{n-1}, 1)\right)^{n_1-n_{10}}, \\
&= \Pr_{i\to i-n}(0 \mid w_{n-1}, 1) \times \Bigg[ \sum_{n_1=1}^{n_{w_{n-1}}-1} \Pr\left(N_{1,w_{n-1}} = n_1\right) \\
&\quad \times \underbrace{\sum_{n_{10}=1}^{n_1} \binom{n_1-1}{n_{10}-1} \Pr_{i\to i-n}(0 \mid w_{n-1}, 1)^{n_{10}-1} \left(1 - \Pr_{i\to i-n}(0 \mid w_{n-1}, 1)\right)^{n_1-n_{10}}}_{=1} \Bigg] \\
&= \Pr_{i\to i-n}(0 \mid w_{n-1}, 1), \\
&= \Pr_{i\to i-n+1}(0 \mid w_{n-1}) \times (1 - \delta_{n,n}).
\end{aligned}$$

And similarly, for the second expected value, we prove that:

$$\mathbb{E}\left[\frac{N_{0,w_{n-1},1}}{N_{0,w_{n-1}}}\right] = \Pr_{i\to i-n+1}(1 \mid w_{n-1}) \times (1 - \delta_{n,n}).$$

Thus, we obtain as announced:

$$\begin{aligned}
\mathbb{E}\left[\mathscr{F}_n^\star \mid w_{n-1}\right] &= 1 - \left[\Pr_{i\to i-n}(0 \mid w_{n-1}) + \Pr_{i\to i-n}(1 \mid w_{n-1})\right] \times (1 - \delta_{n,n}), \\
&= \delta_{n,n}.
\end{aligned}$$

□

We therefore have a statistic to estimate the relevant term in our model from the data of the analyzed sequence. However, the main drawback of this statistic is the need to observe a given word $w_{n-1} \in \{0,1\}^{n-1}$ in the sequence to obtain the estimation of $\delta_{n,n}$. To obtain an estimation of good quality, it is desirable to have a large number of occurrences of $w_{n-1}$ in the sequence, but, on average, this number of occurrences decreases exponentially with $n$. More precisely, on average, this statistic uses only $\frac{N}{2^{n-1}}$ bits of the sequence.

However, as this statistic is defined for all $w_{n-1} \in \{0,1\}^{n-1}$ (as long as the word is preceded at least once by a bit 0 and by a bit 1), it seems possible to develop another statistic which estimates $\delta_{n,n}$ using all $(n-1)$-bit words encountered in the sequence. To this end, we propose the following statistic:

DEFINITION 9.10 (Finer autocorrelation statistic). *For $n > 1$, and a sequence $\mathscr{S}$, we define the **finer autocorrelation statistic** as:*

$$\mathscr{F}_n^\star = \frac{1}{N-n+1} \sum_{\substack{w_{n-1} \in \{0,1\}^{n-1} \\ w_{n-1} \in \mathscr{S}}} N_{w_{n-1}} \times \left(1 - \frac{N_{1,w_{n-1},0}}{N_{1,w_{n-1}}} - \frac{N_{0,w_{n-1},1}}{N_{0,w_{n-1}}}\right).$$

*where $N$ is the number of bits in the sequence $\mathscr{S}$ and $N_{w_{n-1}}$ is the number of occurrences of the word $w_{n-1}$ in $\mathscr{S}$.*

With the statistic defined above, we have the following theorem:

THEOREM 9.11. *For $n \in \mathbb{N}, n > 1$, and $\mathscr{F}_n^\star$ the finer autocorrelation statistic established in Def. 9.10:*

$$\mathbb{E}\left[\mathscr{F}_n^\star\right] = \delta_{n,n}.$$

*In other words, $\mathscr{F}_n^\star$ is an unbiased estimator of $\delta_{n,n}$.*

PROOF. In all generality:

$$\mathbb{E}\left[\mathscr{F}_n^\star\right] = \frac{1}{N-n+1} \times \sum_{\substack{w_{n-1} \in \{0,1\}^{n-1} \\ w_{n-1} \in \mathscr{S}}} \left(\mathbb{E}\left[N_{w_{n-1}}\right] - \mathbb{E}\left[N_{w_{n-1}} \frac{N_{1,w_{n-1},0}}{N_{1,w_{n-1}}}\right] - \mathbb{E}\left[N_{w_{n-1}} \frac{N_{0,w_{n-1},1}}{N_{0,w_{n-1}}}\right]\right).$$

More specifically:

$$\begin{aligned}
\mathbb{E}\left[N_{w_{n-1}}\frac{N_{1,w_{n-1},0}}{N_{1,w_{n-1}}}\right] = & \sum_{n_{w_{n-1}}=1}^{N-n+1}\sum_{n_1=1}^{n_{w_{n-1}}-1}\sum_{n_{10}}^{n_1} n_{w_{n-1}}\frac{n_{10}}{n_1} \\
& \times \Pr\left(N_{1,w_{n-1},0}=n_{10}, N_{1,w_{n-1}}=n_1, N_{w_{n-1}}=n_{w_{n-1}}\right), \\
= & \Pr_{i\to i-k}(0 \mid w_{n-1},1)\times\sum_{n_{w_{n-1}}=1}^{N-n+1} n_{w_{n-1}} \\
& \times \underbrace{\sum_{n_1=1}^{n_{w_{n-1}}-1}\Pr\left(N_{1,w_{n-1}}=n_1, N_{w_{n-1}}=n_{w_{n-1}}\right)}_{=\Pr\left(N_{w_{n-1}}=n_{w_{n-1}}\right)}, \\
= & \Pr_{i\to i-n}(0 \mid w_{n-1},1)\times\mathbb{E}\left[N_{w_{n-1}}\right], \\
= & (1-\delta_{n,n})\times\Pr_{i\to i-n+1}(0 \mid w_{n-1})\times\mathbb{E}\left[N_{w_{n-1}}\right].
\end{aligned}$$

And, similarly:

$$\mathbb{E}\left[N_{w_{n-1}}\frac{N_{0,w_{n-1},1}}{N_{0,w_{n-1}}}\right] = (1-\delta_{n,n})\times\Pr_{i\to i-n+1}(1 \mid w_{n-1})\times\mathbb{E}\left[N_{w_{n-1}}\right].$$

Thus, we finally have:

$$\begin{aligned}
\mathbb{E}\left[\mathscr{F}_n^\star\right] &= \frac{1}{N-n+1}\times\sum_{\substack{w_{n-1}\in\{0,1\}^{n-1}\\ w_{n-1}\in\mathscr{S}}}\mathbb{E}\left[N_{w_{n-1}}\right]\times(1-(1-\delta_{n,n})), \\
&= \delta_{n,n}\times\frac{1}{N-n+1}\times\mathbb{E}\left[\sum_{\substack{w_{n-1}\in\{0,1\}^{n-1}\\ w_{n-1}\in\mathscr{S}}}N_{w_{n-1}}\right].
\end{aligned}$$

And, by construction:

$$\sum_{\substack{w_{n-1}\in\{0,1\}^{k-1}\\ w_{n-1}\in\mathscr{S}}}N_{w_{n-1}} = N-n+1.$$

Therefore, we have, as announced:

$$\mathbb{E}\left[\mathscr{F}_n^\star\right] = \delta_{n,n}.$$

□

We have thus defined above a statistic taking into account all the bits of the sequence (or more precisely, all the $(n-1)$-bit words encountered), which is an unbiased estimator of $\delta_{n,n}$ and

thus, according to our definition, enables us to define an optimal test for a fine-grained analysis of correlations.

However, this new statistic also has a major drawback, which is its cost in both memory and computation time. Indeed, since the aim is to take advantage of all of the $(n-1)$-bit words in the analyzed sequence to estimate $\delta_{n,n}$ as precisely as possible, it is of course necessary to store for each $w_{n-1} \in \{0,1\}$, every counter $N_{w_{n-1}}$, $N_{1,w_{n-1}}$, $N_{0,w_{n-1}}$, $N_{1,w_{n-1},0}$ and $N_{0,w_{n-1},1}$ used to compute $\mathscr{F}_n^\star$. The **memory complexity** of the computation of $\mathscr{F}_n^\star$ is then equal to $\boldsymbol{O(2^n)}$. Also, the estimation is done by browsing once through the whole sequence of length $N$ to increment the aforementioned counters, before summing them as per the formula of the statistic. This then leads to a **computation time complexity** of $\boldsymbol{O(N\ 2^n)}$. In this sense, we limited ourselves to $n = 32$.

Since we have limited ourselves to $N = 100{,}000$ bits of data for the sequences, in the following subsection we will illustrate only the total statistic defined in Def. 9.10, and not the statistic given $w_{n-1}$ defined in Def. 9.8, as the number of occurrences of a specific $(n-1)$-bit word will almost surely becomes zero as $n$ increases.

## 9.3. Applying the fine autocorrelation statistic on simulated sequences

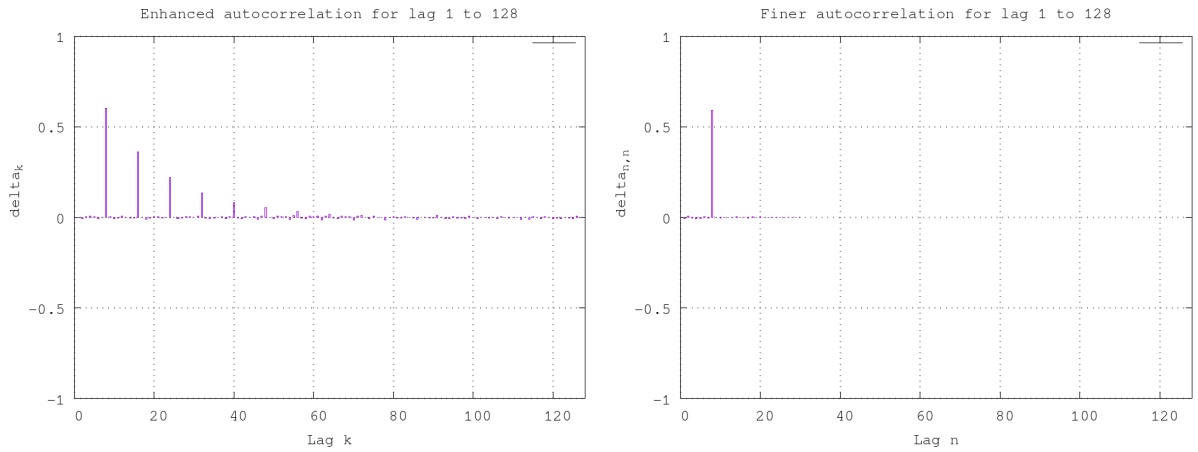
Once again, we sought to validate our statistic by applying it to datasets impacted by correlation phenomena. More specifically, our goal was to demonstrate the interest of this new statistic in the face of the enhanced autocorrelation statistic established in Def. 5.2, we reused the same datasets as the ones used in subsections 7.3.2 and 7.3.3, produced according to the algorithm 5.

In the following figures, the graphs on the right represent the computation of the statistic $\mathscr{F}_n^\star$, which depicts the estimations of the successive $\delta_{n,n}$ terms, when $n$ varies between 1 and 32. For the index $n = 1$, we naturally define the value of $\mathscr{F}_n^\star$ as being equal to the enhanced autocorrelation statistic for a lag 1, i.e. $\mathscr{A}_1^\star$.

**9.3.1. Simulated sequences with known frequency and correlation anomalies.** First of all, we compared the results obtained in the case where a single correlation anomaly between bits distant of 8 affects the sequence.

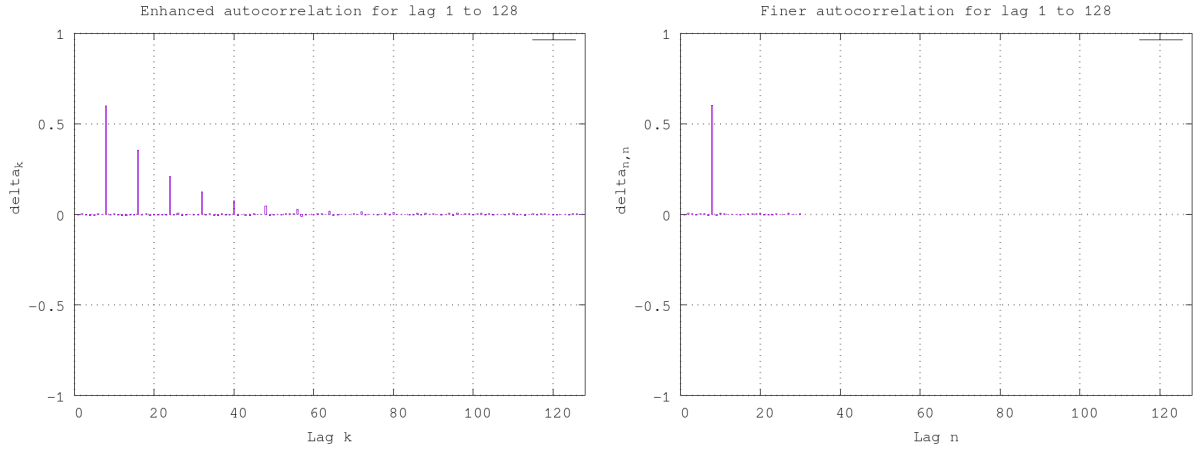
As anticipated and illustrated in Fig. 32, our finer autocorrelation statistic isolates the original correlation phenomenon (between bits distant of 8), and is perfectly null for other lags, in particular for multiples of 8, for which harmonic phenomena were visible with the enhanced autocorrelation statistic $\mathscr{A}_k^\star$. The amplitude of the correlation anomaly between bits distant of 8 is, moreover, rigorously identical to that obtained with the enhanced statistic $\mathscr{A}_k^\star$, which was also expected from our model.

We then applied our statistic $\mathscr{F}_n^\star$ to the dataset affected by the same correlation anomaly between bits distant of 8, as well as by a global disproportion between bits 0 and 1 of amplitude




(A) Enhanced autocorrelation statistic (B) Finer autocorrelation statistic

FIGURE 32. Enhanced autocorrelation statistic $\mathscr{A}_k^\star$ vs Finer autocorrelation statistic $\mathscr{F}_n^\star$, N = 100,000, $\xi = 0$, $\delta_8 = 0.6$

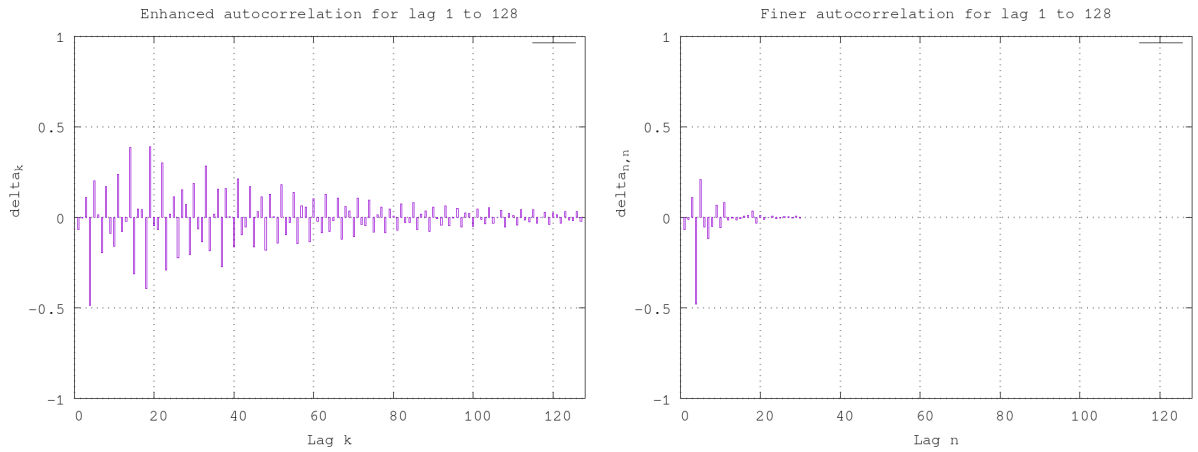



(A) Enhanced autocorrelation statistic (B) Finer autocorrelation statistic

FIGURE 33. Enhanced autocorrelation statistic $\mathscr{A}_k^\star$ vs Finer autocorrelation statistic $\mathscr{F}_n^\star$, N = 100,000, $\xi = 0.6$, $\delta_8 = 0.6$

$\xi = 0.6$.

The result, illustrated in Fig. 33, is once again perfectly in line with our expectations, namely in the fact that the correlation phenomenon between bits distant of 8 is once again the only one identified by the statistic $\mathscr{F}_n^\star$, and that it is in no way affected by the global disproportion




(A) Enhanced autocorrelation statistic (B) Finer autocorrelation statistic

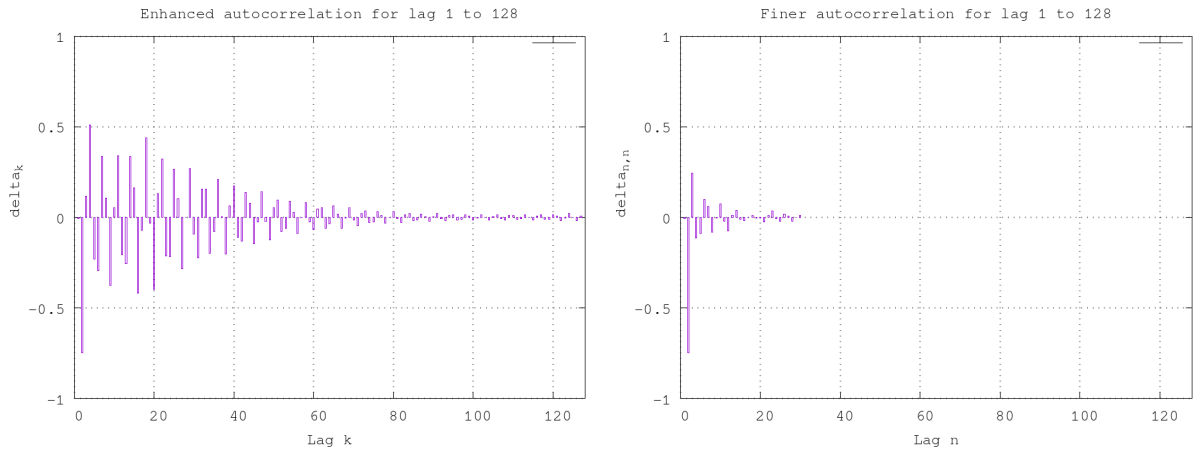
FIGURE 34. Enhanced autocorrelation statistic $\mathscr{A}_k^\star$ vs Finer autocorrelation statistic $\mathscr{F}_n^\star$, ERO-TRNG, $D = 53$

of bits 0 and 1. In this sense, this new statistic does indeed seem to provide a finer analysis of correlations as expected, without any loss in the quality of characterization of actual anomalies.

**9.3.2. Sequences generated by a TRNG implemented on FPGA.** To evaluate the effectiveness of our new statistic in a more concrete use case, we have also applied it to datasets generated using the FPGA implementation of the ERO-TRNG, already used in subsection 7.3.3. Once again, we have deliberately chosen parameters for the generator that lead to a randomness of poor quality in order to be able to easily visualize anomalies, our objective being to judge the ability of our statistics to characterize them. More precisely, we took the parameters $Inv_1 = 11$ and $Inv_2 = 7$, and made $D$ vary between 50 and 100. As a reminder, the number of bits generated for each set of parameters is $N = 114{,}688$ bits.

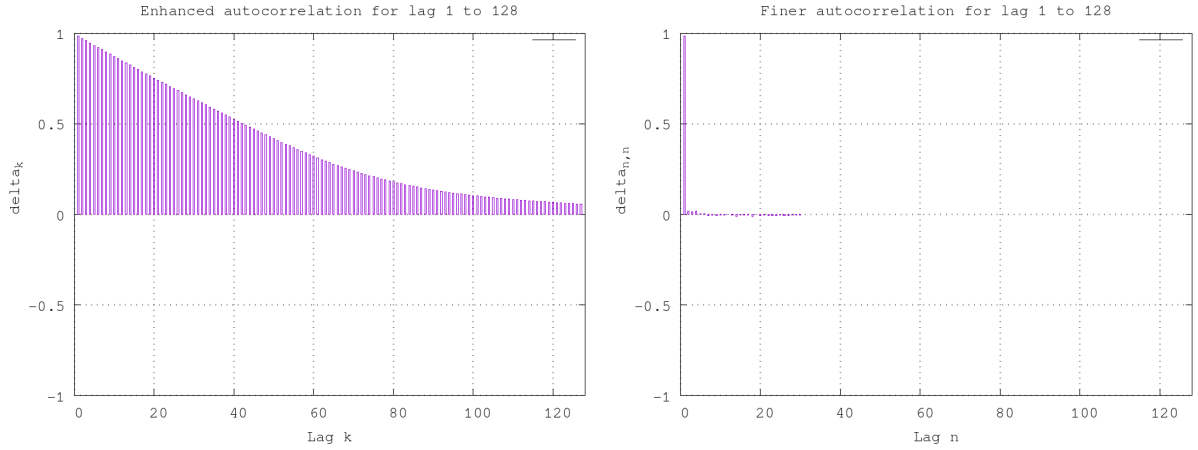
For $D = 53$, for example (see Fig. 34), although the impact of the finer autocorrelation statistic $\mathscr{F}_n^\star$ in the face of the enhanced autocorrelation statistic $\mathscr{A}_k^\star$ is not as obvious as in the case of a single correlation anomaly as presented in the previous subsection, one observation we can make is that $\mathscr{F}_n^\star$ seems to work as a low-pass filter. The sharp cutoff after index $n = 11$ would therefore indicate that the poor parameterization of the generator leads to real correlations extending up to bits distant of 11, but that the rest of the anomalies observed with the enhanced autocorrelation statistic are merely artificial information, despite their high amplitude.

For $D = 84$ (Fig. 35), the results are very similar, with once again a "low-pass" effect of the finer autocorrelation statistic, and real correlations phenomena present up to bits distant of 11 as




(A) Enhanced autocorrelation statistic (B) Finer autocorrelation statistic

FIGURE 35. Enhanced autocorrelation statistic $\mathscr{A}_k^\star$ vs Finer autocorrelation statistic $\mathscr{F}_n^\star$, ERO-TRNG, $D = 84$




(A) Enhanced autocorrelation statistic (B) Finer autocorrelation statistic

FIGURE 36. Enhanced autocorrelation statistic $\mathscr{A}_k^\star$ vs Finer autocorrelation statistic $\mathscr{F}_n^\star$, ERO-TRNG, $D = 55$

well.

Finally, for a value of $D = 55$ (Fig. 36), the impact of the fine autocorrelation statistic is remarkable. Indeed, we had already observed in subsection 7.3.3 that this parameter led to a kind of resonance phenomenon, resulting in an extreme amplitude of the correlation between successive bits. Application of the finer autocorrelation statistic reveals that this correlation phenomenon is in fact limited to bits distant of 1, proving at the same time that the "dummy"

example presented in subsections 7.3.2 and 9.3.1 has a practical reality.

Thus, all these simulations confirm the interest of a finer modeling of correlations, especially from the point of view of a designer who seeks to trace back to the origin of anomalies, by getting rid of artificial correlation information, without deteriorating the characterization of real correlation phenomena.

However, given its computational and memory complexities, this statistic, and therefore our finer autocorrelation model, seems to have limited practical interest, as we explained earlier. We then explored the use of a second correlation model taking into account the influence of $n$ preceding bits, called the *partial autocorrelation*.

## 9.4. Partial autocorrelation function (PACF)

The partial autocorrelation is a concept formalized in particular by Yule (one of the fathers of the theory of correlation with Pearson[1]), who, in 1907 [**82**], introduced notations that would later enable him to theorize the analysis of correlations between the realizations of a given process [**83**].

More precisely, Yule proposes to study the autocorrelation of a stochastic discrete process $\{X_i\}$ by modeling it as an autoregressive process, i.e. a process that verifies an equation like the following:

$$X_i = \phi_{1,n} X_{i-1} + \phi_{2,n} X_{i-2} + ... + \phi_{n,n} X_{i-n},$$

where $n \in \mathbb{N}, n > 0$ is the order of the regression, and $\phi_{1,n}, \phi_{2,n}, ..., \phi_{n,n}$ are the coefficients of the autoregression of order $n$.

Walker [**78**] subsequently showed that, under the hypothesis of this model, a strictly analogous equation subsists between the correlation coefficients, namely:

$$\rho_i = \phi_{1,n} \rho_{i-1} + \phi_{2,n} \rho_{i-2} + ... + \phi_{n,n} \rho_{i-n},$$

where $\phi_{1,n}, \phi_{2,n}, ..., \phi_{n,n}$ are the same coefficients found in the previous equation, and $\rho_i$ is the coefficient of theoretical autocorrelation of order $i$ as defined at the beginning of the section 5.2.

The hypothesis of stationarity also provides the property $\rho_j = \rho_{-j}$, which leads to the following set of $n$ equations:

$$\rho_j = \phi_{1,n} \rho_{|j-1|} + \phi_{2,n} \rho_{|j-2|} + ... + \phi_{n,n} \rho_{|j-n|},$$

[1] In fact, Yule and Pearson engaged in a real scientific battle on the subject for many years from the end of the nineteenth century to the middle of the twentieth century [**22**].

for, $j \in \{1, ..., n\}$, which can also be written as the following matrix equation:

$$\begin{bmatrix} 1 & \rho_1 & \cdots & \cdots & \rho_{n-2} & \rho_{n-1} \\ \rho_1 & 1 & \rho_1 & \cdots & \rho_{n-3} & \rho_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho_{n-2} & \rho_{n-3} & \cdots & \cdots & 1 & \rho_1 \\ \rho_{n-1} & \rho_{n-2} & \cdots & \cdots & \rho_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_{1,n} \\ \phi_{2,n} \\ \vdots \\ \phi_{n-1,n} \\ \phi_{n,n} \end{bmatrix} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{n-1} \\ \rho_n \end{bmatrix}.$$

In our case, we had seen that the coefficients $\rho_i$ are exactly equal to the coefficients $\delta_i$ that we introduced with our simple correlation model (Def. 5.2), for which we have a very efficient estimator. The set of Yule-Walker equations presented above thus translates into:

$$\begin{bmatrix} 1 & \delta_1 & \cdots & \cdots & \delta_{n-2} & \delta_{n-1} \\ \delta_1 & 1 & \delta_1 & \cdots & \delta_{n-3} & \delta_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_{n-2} & \delta_{n-3} & \cdots & \cdots & 1 & \delta_1 \\ \delta_{n-1} & \delta_{n-2} & \cdots & \cdots & \delta_1 & 1 \end{bmatrix} \begin{bmatrix} \phi_{1,n} \\ \phi_{2,n} \\ \vdots \\ \phi_{n-1,n} \\ \phi_{n,n} \end{bmatrix} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{n-1} \\ \delta_n \end{bmatrix}.$$

It is interesting to note that, for $j = n$, the Yule-Walker equation is the following:

$$\delta_n = \phi_{1,n}\delta_{n-1} + \phi_{2,n}\delta_{n-2} + ... + \phi_{n-1,n}\delta_1 + \phi_{n,n},$$

because $\delta_0 = 1$ by construction.

This equation again highlights the linear influence of $\delta_j, 1 \leqslant j \leqslant n-1$ on $\delta_n$, as described by the model. But, above all, it illustrates the importance of the term $\phi_{n,n}$ in characterizing the "real" correlation between bits distant of $n > 0$. Indeed, this term clearly appears as the residual correlation phenomenon when we have removed the influence of all correlation phenomena propagated between realizations distant of $j < n$, in the same way that, in our fine correlation model (see Def. 9.5), the term $(1 - \delta_{n,n})$ was added to the influence of the other correlation terms in the expression of conditional probabilities.

Based on this observation, and building on the work of Yule and Walker, Box and Jenkins [**12**] introduced the *partial autocorrelation function* in 1970.

DEFINITION 9.12 (Partial autocorrelation function). *Let $\phi_{n,n}$ be the $n$-th partial autocorrelation coefficient of order $n$ in the Yule-Walker equation as shown above. We call partial autocorrelation function (abbreviated in PACF) the function:*

$$\begin{array}{rccc} PACF: & \mathbb{N} \setminus \{0\} & \longrightarrow & [-1, 1] \\ & n & \longmapsto & \phi_{n,n} \end{array}.$$

As mentioned earlier, the authors then explain that, as modeled above, the coefficient $\phi_{n,n}$ represents the correlation between $x_i$ and $x_{i-n}$, adjusted to remove the propagation of lower-order correlations (between $x_i$ and $x_{i-j}$, $1 \leqslant j \leqslant n-1$). It thus seems that this function is the perfect tool in our case, and we then want to find a way to estimate $\phi_{n,n}$.

From the set of $n$ Yule-Walker matrix equations, it follows, by Cramer's rule, that the coefficient $\phi_{n,n}$ is obtained as the quotient $d_1^{(n)}/d_2^{(n)}$, where $d_1^{(n)}$ and $d_2^{(n)}$ are the following two determinants:

$$d_1^{(n)} = \begin{vmatrix} 1 & \delta_1 & \cdots & \cdots & \delta_{n-2} & \delta_1 \\ \delta_1 & 1 & \delta_1 & \cdots & \delta_{n-3} & \delta_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_{n-2} & \delta_{n-3} & \cdots & \cdots & 1 & \delta_{n-1} \\ \delta_{n-1} & \delta_{n-2} & \cdots & \cdots & \delta_1 & \delta_n \end{vmatrix} \qquad d_2^{(n)} = \begin{vmatrix} 1 & \delta_1 & \cdots & \cdots & \delta_{n-2} & \delta_{n-1} \\ \delta_1 & 1 & \delta_1 & \cdots & \delta_{n-3} & \delta_{n-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_{n-2} & \delta_{n-3} & \cdots & \cdots & 1 & \delta_1 \\ \delta_{n-1} & \delta_{n-2} & \cdots & \cdots & \delta_1 & 1 \end{vmatrix}$$

The following theorem further convinces us of the usefulness of partial autocorrelation in our use case:

THEOREM 9.13. *For a discrete stationary stochastic process affected by a unique correlation phenomenon between bits distant of $k > 0$, as described by the model established in Def. 5.2, the partial autocorrelation function of order $u > 0$ is equal to:*

$$\phi_{u,u} = \begin{cases} \delta_k, & \text{if } u = k, \\ 0 & \text{if } u \neq k. \end{cases}$$

PROOF. In the event that $\boldsymbol{u = k}$, the result is immediate since, with our model, a single correlation phenomenon between bits distant of $k > 0$ results in $\delta_j = 0$ for $1 \leqslant j \leqslant k-1$ and $\delta_k \neq 0$. Then, $d_1^{(u)}$ is the determinant of the matrix $\text{diag}(1, ..., 1, \delta_k)$ and $d_2^{(u)}$ is the determinant of the identity matrix. We therefore have $\phi_{u,u} = \delta_k$ as announced.

Similarly, in the case where $\boldsymbol{u < k}$, $d_1^{(u)}$ is the determinant of the matrix
$\text{diag}(1, ..., 1, \delta_u) = \text{diag}(1, ..., 1, 0)$, so $d_1^{(u)} = 0$ and $\phi_{u,u} = 0$.

Finally, when $\boldsymbol{u > k}$, let $m \in \mathbb{N}, m > 0$ and $r \in \{1, ..., k-1\}$ such that $u = mk + r$. For $(i, j) \in \{1, ..., u\}^2$, we denote by $d_1^{(u)}(i, j)$ the coefficient of row $i$ and column $j$ of the determinant $d_1^{(u)}$. Remembering that, for $j < u$, $d_1^{(u)}(i, j) = \delta_{|i-j|}$ and $d_1^{(u)}(i, u) = \delta_i$, we have the following relationship on the coefficients of $d_1^{(u)}$:

$$d_1^{(u)}(u,j) = \begin{cases} \delta_{(p-m)k} = \delta_{(m-p)k}, & \text{if } j = pk+r \text{ where } p \in \{0,...,m-1\}, \\ \delta_u, & \text{if } j = u, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$d_1^{(u)}(u-k,j) = \begin{cases} \delta_{(p-m+1)k} = \delta_{(m-1-p)k}, & \text{if } j = pk+r \text{ where } p \in \{0,...,m-1\}, \\ \delta_{u-k}, & \text{if } j = u, \\ 0 & \text{otherwise,} \end{cases}$$

From Th. 9.2, it follows that:

$$\delta_{(m-p)k} = \delta_k^{m-p},$$

$$\delta_{(m-1-p)k} = \delta_k^{m-1-p},$$

and therefore

$$\delta_u = \begin{cases} \delta_{mk} = \delta_k^m, & \text{if } u = mk, \\ 0 & \text{otherwise,} \end{cases}$$

$$\delta_{u-k} = \begin{cases} \delta_{(m-1)k} = \delta_k^{m-1}, & \text{if } u = mk, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, for all $u > k$ and all $j \in \{1,...,u\}$, it follows that $d_1^{(u)}(u,j) = \delta_k \times d_1^{(u)}(u-k,j)$. Since the determinant $d_1^{(u)}$ has two proportional rows, we have $d_1^{(u)} = 0$ and therefore $\phi_{u,u} = 0$ as announced.

□

The partial autocorrelation function thus acts exactly like our finer autocorrelation statistic, completely eliminating artificial correlation phenomena in the case of a single real phenomenon between bits distant of $k > 0$.

Furthermore, the computational and memory complexity is also much better. Indeed, to compute the coefficient $\phi_{n,n}$, the first step is to compute all the terms $\delta_j, 1 \leqslant j \leqslant n$, which is achieved with our enhanced autocorrelation statistic (see section 7.1) with **computational complexity** $O(N \times n)$ and **constant memory complexity**. The second step is then the computation of the term $\phi_{n,n}$ which can be performed naively in computational complexity $O(n^3)$

| | Computational complexity | Memory complexity |
|---|---|---|
| Computing $\delta_1, ..., \delta_n$ + PACF | $O(Nn + n^2)$ | $O(n^2)$ |
| Finer autocorrelation statistic $\mathscr{F}_n^\star$ | $O(N + 2^n)$ | $O(2^n)$ |

TABLE 2. Complexity of the PACF and of the finer autocorrelation statistic

and **memory complexity $O(n^2)$** (which simply corresponds to the storage of the matrices) with the Gauss-Hordan elimination algorithm. This complexity would then naturally already be much better than the complexity of computing the terms $\delta_{n,n}$ of our finer correlation model. However, an even more efficient algorithm exists for computing $\phi_{n,n}$.

Indeed, it is possible to take advantage of the structure of the two determinants, in particular their symmetry (or quasi-symmetry for $d_1^{(n)}$) to simplify the computation. This is the work carried out by Levinson in 1947 [**50**], which was later improved by Durbin [**23**] to produce the Durbin-Levinson algorithm. The algorithm is a recursive method based on the recursion equation (7) in section 2 of [**23**], and enables the computation of the coefficients of an autoregressive model in **computational complexity $O(n^2)$**, with $n > 1$ the order of the regression.

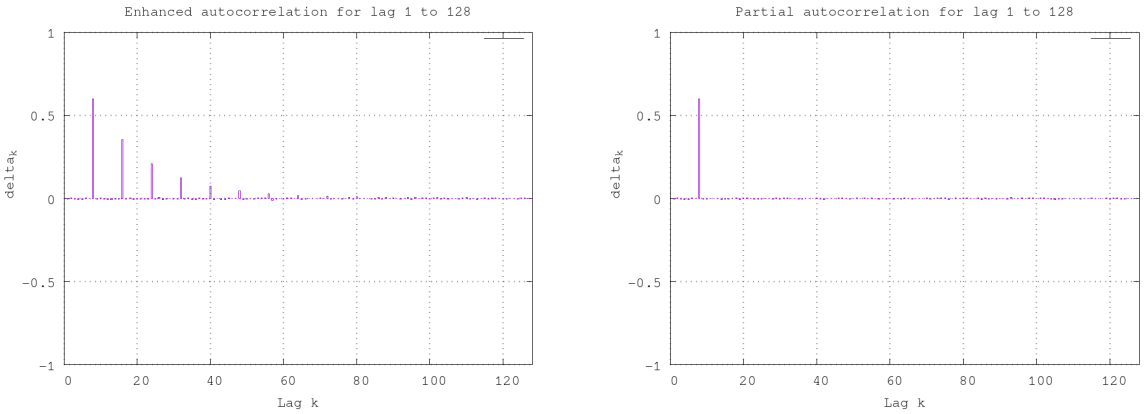
To summarize, the computational and memory complexities of the partial autocorrelation function and the finer autocorrelation statistic are listed in Tab. 2.

## 9.5. Applying the partial autocorrelation function on simulated sequences

It therefore seems that the PACF is able to solve our problematic of precise characterization of real correlation between the bits of a sequence. We then wished to verify the behavior of this function on the same datasets used to illustrate both the enhanced autocorrelation statistic (see section 7.3) and the finer autocorrelation statistic (see section 9.3). The idea being to prove the interest of this new methodology with respect to our enhanced correlation statistic $\mathscr{A}_k^\star$, but also with respect to our finer statistic $\mathscr{F}_n^\star$, we will show for each interesting dataset the result of applying the statistics $\mathscr{A}_k^\star$ and $\mathscr{F}_n^\star$, and the partial autocorrelation function for different orders $n > 0$.

> *Remark :* The Durbin-Levinson algorithm is more efficient than a naive determinant computation, but in our case, we limit our simulations to an order $n = 128$. We have therefore chosen not to implement Durbin-Levinson, as it would not be necessary. Instead, we use the Pari/GP library [**64**] and the GNU MP library [**31**] for optimized determinant computation.

**9.5.1. Simulated sequences with known frequency and correlation anomalies.** First of all, we wanted to validate the result of Th. 9.13, i.e. that, in the case of a single real correlation




(A) Enhanced autocorrelation statistic (B) Partial autocorrelation function

FIGURE 37. Enhanced autocorrelation statistic vs Partial autocorrelation function, N = 100,000, $\xi = 0.6$, $\delta_8 = 0.6$

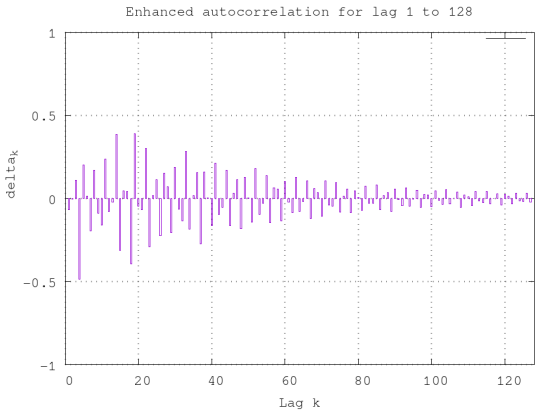
phenomenon between bits distant of $k > 0$, the partial autocorrelation function is indeed able to isolate this phenomenon, and is unaffected by the propagated correlation phenomena.

The application of the partial autocorrelation function to data affected by a correlation phenomenon with parameter $\delta_8 = 0.6$ is then depicted in Fig. 37. The plot of the partial autocorrelation function (on the right) shows on the $x$-axis the order $n > 0$ of the regression considered, and on the $y$-axis the value of the term $\phi_{n,n}$, which therefore represents the "real" correlation between bits distant of $n$.

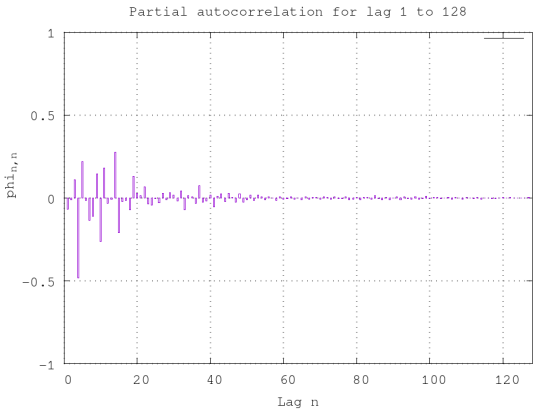
As expected, the autocorrelation function is perfectly able to isolate the real correlation phenomenon between bits distant of 8, and to completely obscure propagated phenomena.

It may also be noted that, for this dataset, the relatively strong overall disproportion of bits 0 and 1 we had set ($\xi = 0.6$) did not impinge on the characterization of the correlation phenomena. This is actually logical, since the term $\phi_{n,n}$ is a function of only $\delta_j, 1 \leqslant j \leqslant n$, whose estimations are independent from $\xi$ (in their expected values).

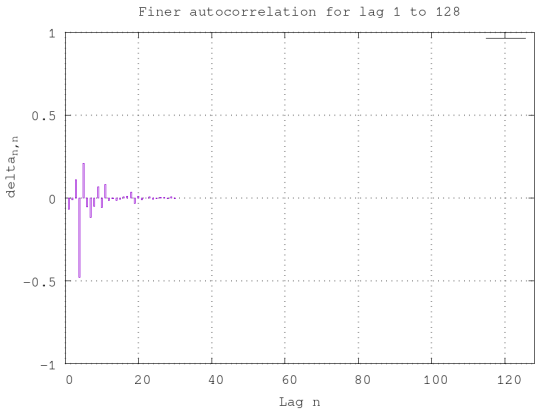
**9.5.2. Sequences generated by a TRNG implemented on FPGA.** Once again, we wanted to test the new methodology, that is the use of the partial autocorrelation function, on more concrete data, i.e. the data sets generated by the FPGA implementation of the ERO-TRNG. More precisely, we have applied the partial autocorrelation function to datasets using the same parameters of the generator as those used in subsection 9.3.2, in order to be able to directly compare the results with each other.




(A) Enhanced autocorrelation statistic, $D = 53$




(B) PACF, $D = 53$

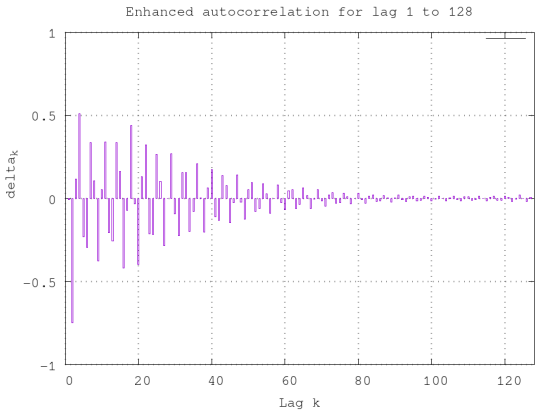



(C) Finer autocorrelation statistic, $D = 53$

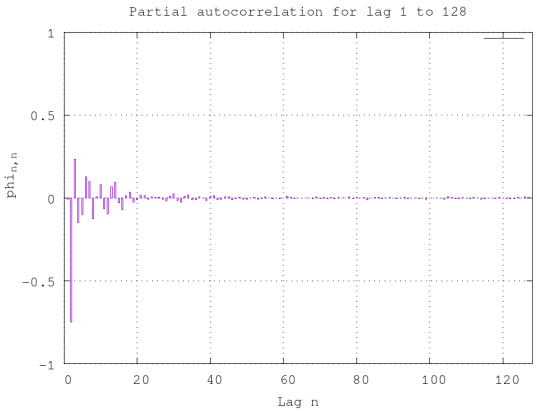
FIGURE 38. Different autocorrelation statistics, ERO-TRNG, D = 53

It appears from the first figure (see Fig. 38), obtained with a frequency division factor $D = 53$, that the use of the partial autocorrelation function leads to very similar results to those obtained with our finer autocorrelation statistic for lags $n \leqslant 7$. On the other hand, the partial autocorrelation function reveals correlation phenomena that our finer autocorrelation statistic had almost totally eliminated, notably for lags $n \geqslant 12$ in this figure. It therefore seems that the partial autocorrelation function is less "aggressive" than our finer autocorrelation statistic, which can be explained by the different nature of the two models: the partial autocorrelation function is based on a linear model, while our statistic is based on a multiplicative one (see Def. 9.5).

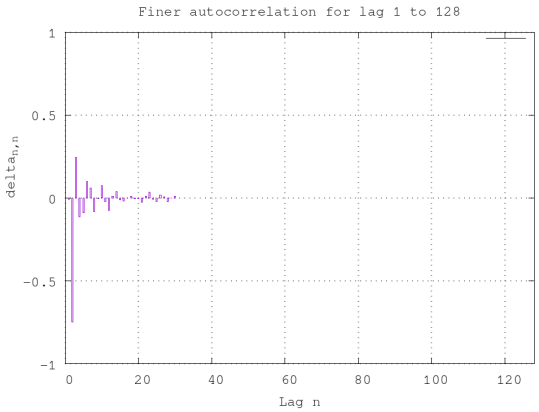
For the sequence generated with the parameter $D = 84$ (Fig. 39), the observation is the same, although this time the difference in the amplitude of the correlations for lags $n \geqslant 12$ is less remarkable.




(A) Enhanced autocorrelation statistic, $D = 84$




(B) PACF, $D = 84$

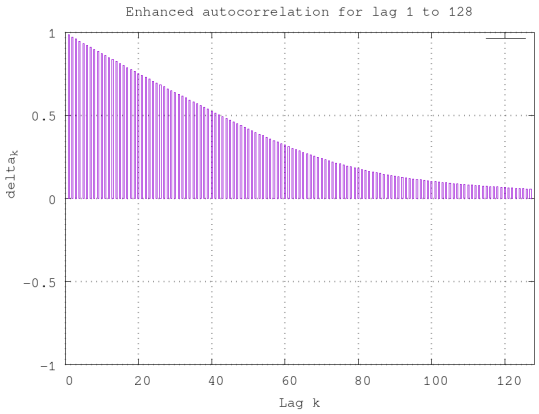



(C) Finer autocorrelation statistic, $D = 84$

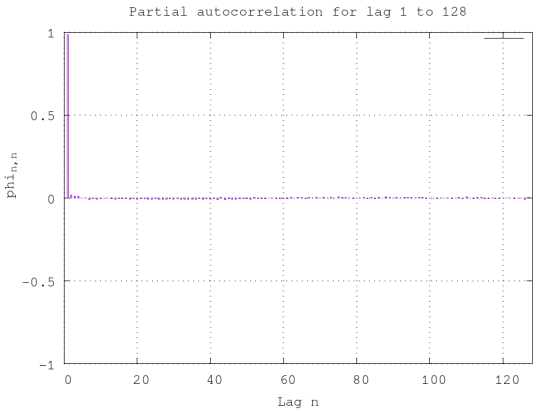
FIGURE 39. Different autocorrelation statistics, ERO-TRNG, D = 84

Finally, for the sequence generated with the parameter $D = 55$ (Fig. 40), the partial autocorrelation function is, like our fine autocorrelation statistic, perfectly able to isolate the unique "real" phenomenon of correlation between bits distant of $k = 1$.

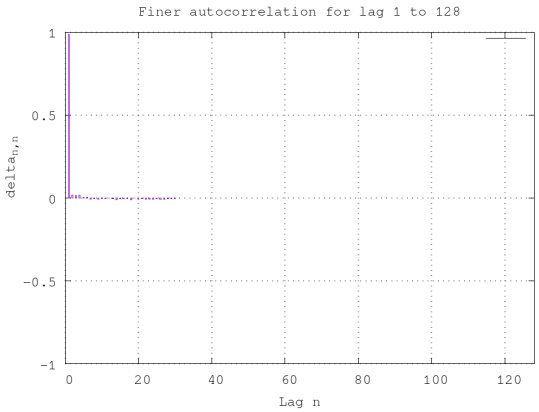
It therefore appears that, in practice, the results provided by the partial autocorrelation function are virtually identical to those provided by our finer autocorrelation statistic. In this sense, and in light of the complexity of both methods (see Tab. 2), this function, combined with a prior computation of the autocorrelation coefficients using our enhanced autocorrelation statistic, seems to be the perfect tool for a precise and relatively inexpensive characterization of the correlation phenomena between bits in a data sequence.




(A) Enhanced autocorrelation statistic, $D = 55$




(B) PACF, $D = 55$




(C) Finer autocorrelation statistic, $D = 55$

FIGURE 40. Different autocorrelation statistics, ERO-TRNG, D = 55

CHAPTER 10

# Conclusion

In this manuscript, we have given an overview of random number generation methods, be they deterministic, non-deterministic or based on physical phenomena, as well as methods to evaluate the quality of the randomness produced by these generators. We then focused on one of the aspects of this analysis of the quality of the randomness, which is the application of black box tests.

In particular, we focused on one of the standard tests called the *autocorrelation test* of the AIS 20/31 standard, developed by the German agency BSI but used in the whole world for the analysis of random number generators. The aim of this test is to detect the presence of potential correlations in the generated data, which should not be present in ideally random data, as they could make future (or past) data predictable.

To analyze this test in greater depth, we have proposed a definition of what the ANSSI guide names (but does voluntarily not define) a "defect in the randomness", which we have chosen to name a *statistical anomaly*. We have then modeled two anomalies, namely correlations between successive bits, and the overall disproportion of bits 0 and 1 in a data sequence.

Using these two models, we then demonstrated that the autocorrelation test of the AIS 20/31 had the drawback of being impacted by anomalies for which it was supposedly not developed, namely the global disproportion of bits 0 and 1, which is the focus of a test of its own. From this observation, we have chosen to develop a new statistic, which would *only* be impacted by an *autocorrelation* between the bits of a sequence. Our newly established correlation model then enabled us to design a very simple statistic to measure the autocorrelation in a binary sequence, that we named *enhanced autocorrelation statistic*. Moreover, our statistic has the advantage of providing a detailed characterization of the correlation phenomena, and not only their detection as the AIS 20/31 test does. In this sense, the statistic we propose enables the definition of a relevant parametric test rather than a mere hypothesis test.

Once our test had been properly proven and developed, we explored another use for it, namely the detection of the success of an attack on a random number generator. An attack has for objective to modify the distribution of the output numbers, thus our test appeared to be the perfect tool to observe the impact of the attack on the produced sequences as an image of its success. Our results on a frequency injection attack show that, when the attack is a success, the impact on the correlations is very visible, making it possible to distinguish the frequency

ranges for which the generator is vulnerable. Using our test also allows for the automation of the detection of the success of an attack, whereas the current methodology requires manually checking the data to verify whether the attack succeeded or not.

Finally, despite the effectiveness of our model and our autocorrelation statistic, one last problem remained, which is that this statistic leads, for each correlation phenomenon, to the appearance of other "artificial" peaks of correlation. From the point of view of a designer of random number generators in particular, it is interesting to be able to isolate the real correlation phenomena, i.e. the phenomena that have a practical origin, in order to identify what needs to be corrected more easily. To that extent, we explored the use of two, more complex, models to characterize the correlations in a binary sequence. The first one is an extension of our previous model, while the second, called *partial autocorrelation*, is an existing model, but is based on a preliminary calculation of coefficients using our aforementioned enhanced autocorrelation statistic. We then saw that the two complex models provide similar and convincing results on the isolation of real correlation phenomena, but that partial autocorrelation is much less expensive in both memory and computation time. Then, the methodology we recommend to evaluate the presence of correlation in a dataset is to use the partial autocorrelation function, with a prior computation of the correlation coefficients with our enhanced autocorrelation statistic.

We have thus set up a methodology to precisely characterize correlations within a binary sequence, based on proven models, which can be used in the context of designing, testing or attacking a random number generator as demonstrated here.

As a follow-up to this work, this methodology, and more precisely our philosophy of a prior modeling of the statistical anomalies can serve as a groundwork to create new test procedures, or enhance existing ones. These new procedures can then be used in a context of evaluation, or to perform an attack as we showcased. In the case of an attack on a generator, new metrics based on correlation factors can be tested in conjunction with, or as a replacement of the metric we proposed to visualize the appearance of correlations in the data. New statistical anomalies could also be explored and modeled to create yet more tests, which would ideally be optimal for the evaluation of a given anomaly, according to our definition of the optimality of a test.

# Bibliography


[1] ANSSI. Guide des mécanismes cryptographiques - règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques. v.2.04.

[2] ASH, R. B. Information theory. *Dover Publication* (1990).

[3] AZIZA, H., POSTEL-PELLERIN, J., BAZZI, H., CANET, P., MOREAU, M., MARCA, V. D., AND HARB, A. True random number generator integration in a resistive ram memory array using input current limitation. *IEEE Transactions on Nanotechnology, vol. 19, pp. 214-222. DOI: 10.1109/TNANO.2020.2976735* (2020).

[4] BARKER, E., AND KELSEY, J. Sp 800-90a. recommendation for random number generation using deterministic random bit generators. *Technical report, Gaithersburg, MD, United States* (2015).

[5] BARKER, E., KELSEY, J., MCKAY, K., ROGINSKY, A., AND TURAN, M. Sp 800-90c. recommendation for random bit generator (rbg) constructions. (3rd public draft). *Technical report, Gaithersburg, MD, United States. https://doi.org/10.6028/NIST.SP.800-90C.3pd* (2022).

[6] BARKER, E., KELSEY, J., TURAN, M., MCKAY, K., BAISH, M., AND M.BOYLE. Sp 800-90b. recommendation for the entropy sources used for random bit generation. *Technical report, Gaithersburg, MD, United States* (2018).

[7] BAUDET, M., LUBICZ, D., MICOLOD, J., AND TASSIAUX, A. On the security of oscillator-based random number generators. *Journal of Cryptology* (2011).

[8] BAYON, P., BOSSUET, L., AUBERT, A., AND FISHER, V. Fault model of electromagnetic attacks targeting ring oscillator-based true random number generators. *Journal of Cryptographic Engineering 6*, 1 (2016), 61–74.

[9] BAYON, P., BOSSUET, L., AUBERT, A., FISHER, V., POUCHERET, F., ROBISSON, B., AND MAURINE, P. Contactless electromagnetic active attack on ring oscillator based true random number generator. *Constructive Side-Channel Analysis and Secure Design 7275* (2012), 151–166.

[10] BERLEKAMP, E. R. Algebraic coding theory. *McGraw-Hill* (1968).

[11] BLUM, M., AND MICALI, S. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing 13*, 4 (1984), 850–864.

[12] BOX, G. E. P., JENKINS, G. M., REINSEL, G. C., AND LJUNG, G. M. Times series analysis : Forecasting and control. fifth edition. *Wiley* (2015).

[13] "BROWN, G. W. *"History of RAND's Random Digits: Summary"*. RAND Corporation, "Santa Monica, CA", "1949".

[14] BROWN, R. G. Dieharder: A random number test suite. *https://webhome.phy.duke.edu/ rgb/General/dieharder.php* (2003).

[15] BROWN, R. G. Dieharder: A gnu public license random number tester. *Technical report* (2006).

[16] BUCHOVECKA, S., AND HLAVAC, J. Frequency injection attack on a random number generator. *2013 IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS). DOI: 10.1109/DDECS.2013.6549803* (2013), 128–130.

[17] CEA-LETI. Opentrng project. `https://opentrng.org`, `https://github.com/opentrng`.

[18] COOK, S. A. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1971), STOC '71, Association for Computing Machinery, p. 151–158.

[19] COOPER, D., SANTESSON, S., FARRELL, S., BOEYEN, S., HOUSLEY, R., AND POLK, W. Rfc 5280: Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile, 2008.

[20] CORON, J.-S., AND NACCACHE, D. An accurate evaluation of maurer's universal test. *Selected Areas in Cryptography, pp. 57–71* (1998).

[21] DE JULIS, G. Analyse d'accumulateurs d'entropie pour les générateurs aléatoires cryptographiques. *Thèse, Mathématiques [math]. ED MSTII, 2014. Français* (2014).

[22] DROESBEKE, J.-J., AND TASSI, P. George udny yule ou comment (ne pas) parler de corrélation. *Statistique et analyse des données 15* (1990), 25–43.

[23] DURBIN, J. The fitting of times series. *Revue de l'Institut International de Statistique / Review of the International Statistical Institute 28*, 3 (1960), 233–244.

[24] DURGA, R. S., RASHMIKA, C. K., MADHUMITHA, O. N. V., SUVETHA, D. G., TANMAI, B., AND MOHANKUMAR, N. Design and synthesis of lfsr based random number generator. *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 438-442. DOI: 10.1109/ICSSIT48917.2020.9214240* (2020).

[25] EBALARD, A., AND BANADJILA, R. Randomness of random in cisco asa. *SSTIC 2023, Rennes* (2023).

[26] ELBAZ-VINCENT, P., AND TRAORÉ, M. Revisiting the pervasiveness of weak keys in network devices. In *2021 IEEE Security and Privacy Workshops (SPW)* (2021), pp. 43–48.

[27] FAILOVERFLOW. Console hacking 2010. ps3 epic fail. *27th Chaos Communication Congress. https://fahrplan.events.ccc.de/congress/2010/Fahrplan/attachments/1780_27c3_console_hacking_2010.pdf* (2010).

[28] FISCHER, V., AND DRUTAROVSKÝ, M. True random number generator embedded in reconfigurable hardware. *Cryptographic Hardware and Embedded Systems - CHES 2002 2523* (2002), 415–430.

[29] FISCHER, V., AND LUBICZ, D. Embedded evaluation of randomness in oscillator based elementary trng. *Workshop on Cryptographic Hardware and Embedded Systems 2014 (CHES 2014)* (2014).

[30] FISHER, R. A., AND YATES, F. Statistical tables for biological, agricultural and medical research. 3rd edition. *Oliver and Boyd, London* (1948).

[31] GMP DEVELOPMENT TEAM. *GNU Multiple Precision Arithmetic Library version 6.3.0*, 2023. available from `https://gmplib.org/`.

[32] GOLDREICH, O., KRAWCZYK, H., AND LUBY, M. On the existence of pseudorandom generators. In *Proceedings on Advances in Cryptology* (Berlin, Heidelberg, 1988), CRYPTO '88, Springer-Verlag, p. 146–162.

[33] HAGERTY, P., AND DRAPER, T. Entropy bounds and statistical tests. *NIST Random Bit Generation Workshop* (2012), 1–28.

[34] HAJIMIRI, A., AND LEE, T. A general theory of phase noise in electrical oscillators. *IEEE Journal of Solid-State Circuits 33*, 2 (1998), 179–194.

[35] HAJIMIRI, A., LIMOTYRAKIS, S., AND LEE, T. Jitter and phase noise in ring oscillators. *IEEE Journal of Solid-State Circuits 34*, 6 (1999), 790–804.

[36] HOND, B., AND HAJIMIRI, A. A general theory of injection locking and pulling in electrical oscillators—part i: Time-synchronous modeling and injection waveform design. *IEEE Journal of Solid-State Circuits, DOI: 10.1109/JSSC.2019.2908753 54*, 8 (2019), 2109–2121.

[37] HOOGE, F. Discussion of recent experiments on $1/f$ noise. *Physica 60*, 1 (1972), 130–144.

[38] HOPCROFT, J. E., MOTWANI, R., AND ULLMAN, J. D. Introduction to automata theory, languages, and computation. *Addison Wesley Longman* (2006).

[39] IMPAGLIAZZO, R., LEVIN, L. A., AND LUBY, M. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1989), STOC '89, Association for Computing Machinery, p. 12–24.

[40] INTEL. Intel® digital random number generator (drng), rev. 2.1. *https://www.intel.com/content/www/us/en/developer/articles/guide/intel-digital-random-number-generator-drng-software-implementation-guide.html* (2018).

[41] JENSEN, J. L. W. V. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Math., 1906, 30: 175-193. DOI: 10.1007/BF02418571* (1906).

[42] KENDALL, M. G., AND SMITH, B. B. Randomness and random sampling numbers. *Journal of the Royal Statistical Society 101*, 1 (1938), 147–166.

[43] KILLMANN, W., AND SCHINDLER, W. A proposal for : Functionality classes for random number generators. *Online. Available at: https://www.bsi.bund.de* (2011).

[44] KNUTH, D. E. On measures of entropy and information. *The art of computer programming. Vol. 2 Seminumerical algorithms. First edition. Addison-Wesley Series in Computer Science and Information Processing* (1969).

[45] L'ECUYER, P. History of uniform random number generation. In *WSC 2017 - Winter Simulation Conference* (Las Vegas, United States, Dec. 2017).

[46] L'ECUYER, P., AND SIMARD., R. Testu01 : A c library for empirical testing of random number generators. *ACM Transactions on Mathematical Software* (2007).

[47] LEHMANN, E. L., AND ROMANO, J. P. Testing statistical hypotheses. 4th ed. *Springer* (2022).

[48] LEINSTER, T. *Entropy and Diversity: The Axiomatic Approach*. Cambridge University Press, 2021.

[49] LEVIN, L. A. One-way functions and pseudorandom generators. STOC '85, Association for Computing Machinery, p. 363–365.

[50] LEVINSON, N. The wiener (root mean square) error criterion in filter design and prediction. *Journal of Mathematics and Physics 25* (1946), 261–278.

[51] LUBICZ, D. On a classification of finite statistical tests. *Advances in Mathematics of Communications, 1(4): 509-524. DOI: 10.3934/amc.2007.1.509* (2007).

[52] MARKETOS, T. A., AND MOORE, S. W. The frequency injection attack on ring-oscillator-based true random number generators. *Cryptographic Hardware and Embedded Systems - CHES 2009 5747* (2009), 317–331.

[53] MARSAGLIA, G. The marsaglia random number cdrom including the diehard battery of tests of randomness. *Technical report. Tests available at : https://ani.stat.fsu.edu/diehard/* (1995).

[54] MARTIN, H., KORAK, T., MILLAN, E. S., AND HUTTER, M. Fault attacks on strngs: Impact of glitches, temperature, and underpowering on randomness. *IEEE Transactions on Information Forensics and Security 10*, 2 (2015), 266–277.

[55] MASSEY, J. Guessing and entropy. 204–.

[56] MASSEY, J. L. Shift-register synthesis and bch decoding. *IEEE Transactions on Information Theory, vol. 15, pp. 122–127* (1969).

[57] MAURER, U. M. A universal statistic test for random bit generators. *Journal of Cryptology, Volume 5, no. 2, pp. 89-105* (1992).

[58] MCSHANE, E. A., AND SHENAI, K. The electrical engineering handbook : 2 - noise in analog and digital systems. *DOI: 10.1016/B978-012170960-0/50011-6* (2005), 101–108.

[59] MENEZES, A., VAN OORSCHOT, P., AND VANSTONE, S. Chapter 5 : Pseudorandom bits and sequences. *Handbook of Applied Cryptography* (1997), 169–190.

[60] NGUYEN, P. Q., AND SHPARLINSKI, I. E. The insecurity of the digital signature algorithm with partially known nonces. *Journal of Cryptography 15*, 3 (2002), 151–176.

[61] NGUYEN, P. Q., AND SHPARLINSKI, I. E. The insecurity of the elliptic curve digital signature algorithm with partially known nonces. *Designs, Codes and Cryptography 30* (2003), 201–217.

[62] NYQUIST, H. Thermal agitation of electric charge in conductors. *Phys. Rev. 32* (Jul 1928), 110–113.

[63] OF STANDARDS, N. I., AND TECHNOLOGY. Fips pub 140-2 : Security requirements for cryptographic modules. *Technical report, Gaithersburg, MD, United States.* (2001).

[64] PARI GROUP. *PARI/GP version 2.13.4*. Univ. Bordeaux, 2022. available from `http://pari.math.u-bordeaux.fr/`.

[65] PARK, K. I. Fundamentals of probability and stochastic processes with applications to communications. *Springer International Publishing. DOI: 10.1007/978-3-319-68075-0* (2018).

[66] PEBAY-PEYROULA, F., DALGATY, T., AND VIANELLO, E. Entropy source characterization in hfo2 rram for trng applications. *15th Design and Technology of Integrated Systems in Nanoscale Era (DTIS). DOI: 10.1109/DTIS48698.2020.9081294* (2020).

[67] PETER, M., AND SCHINDLER, W. A proposal for : Functionality classes for random number generators (draft sept. 2022). *Online. Available at: https://www.bsi.bund.de* (2022).

[68] PETURA, O. True random number generators for cryptography: Design, securing and evaluation. *Thèse. Micro and nanotechnologies/Microelectronics. English* (2019).

[69] RENYI, A. On measures of entropy and information. *Berkeley Symp. on Math. Statist. and Prob., pp 547-561* (1961).

[70] RUKHIN, A., SOTO, J., NECHVATAL, J., SMID, M., BARKER, E., LEIGH, S., LEVENSON, M., VANGEL, M., BANKS, D., HECKERT, A., DRAY, J., AND VO, S. Sp 800-22. a statistical test suite for random and pseudorandom number generators for cryptographic applications. *Technical report, Gaithersburg, MD, United States* (2010).

[71] SA, I. Q. Random number generation: What is the q in qrng? *White Paper* (2020).

[72] SARALEES, N., AND SAMUEL, K. Computation of signal to noise ratios. *MATCH - Communications in Mathematical and in Computer Chemistry 57* (2007), 105–110.

[73] SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal, vol.27, pp 379-423* (1948).

[74] SHRIMPTON, T., AND TERASHIMA, R. S. A provable-security analysis of intel's secure key rng. In *Advances in Cryptology – EUROCRYPT 2015* (2015), Springer Berlin Heidelberg, pp. 77–100.

[75] SOUCARROS, M., CLÉDIÈRE, J., DUMAS, C., AND ELBAZ-VINCENT, P. Fault analysis and evaluation of a true random number generator embedded in a processor. *Journal of Electronic Testing 29*, 3 (2013), 367–381.

[76] SPARAVIGNA, A. C. Poissonian distributions in physics: Counting electrons and photons. *hal-03126250* (2021).

[77] TRAORE, M. *Analyse des biais de RNG pour les mécanismes cryptographiques et applications industrielles.* Thèse, Université Grenoble Alpes , May 2022.

[78] WALKER, G. On periodicity in series of related terms. *Proc. R. Soc. Lond. A 131* (1931), 518–532.

[79] WOLD, K., AND PETROVI, S. Robustness of trng against attacks that employ superimposing signal on fpga supply voltage. *Proceedings of the Norwergian information security conference, NISK* (2010), 81–92.

[80] YAO, A. C. Theory and applications of trapdoor functions. *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982), IEEE, pp. 80–91. DOI: 10.1109/SFCS.1982.45* (1982).

[81] YEE, P. Rfc 6818: Updates to the internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile, 2013.

[82] YULE, G. U. On the theory of correlation for any number of variables treated by a new system of notation. *Proc. R. Soc. Lond. A 79* (1907), 182–193.

[83] YULE, G. U. On a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. *Proc. R. Soc. Lond. A 226* (1927), 267–298.