A New Statistical Test for PRNG Based on the Attendance’s Law

Babacar Alasane Ndaw¹,²*, Ousmane Ndiaye², Mamadou Sangharé² and Cheikh Thiécoumba Gueye²

¹ Central Technical Service of Cipher and Information Security Systems, Presidency of the Republic, Dakar, Senegal.
² Université Cheikh Anta Diop de Dakar, FST, DMI, LACGAA, Senegal.

Authors’ contributions
This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2021/v36i130328
Editor(s):
(1) Raducanu Razvan, Al. I. Cuza University, Romania.
Reviewers:
(1) Suman De, India.
(2) C. John Moses, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, India.
(3) Pawan Kumar Goel, Shri Ram Group Of Colleges, India.
Complete Peer review History: http://www.sdiarticle4.com/review-history/63674

Received: 03 December 2020
Accepted: 09 February 2021
Published: 10 March 2021

Original Research Article

Abstract

One family of the cryptographic primitives is random Number Generators (RNG) which have several applications in cryptography such that password generation, nonce generation, Initialisation vector for Stream Cipher, keystream. Recently they are also used to randomise encryption and signature schemes.

A pseudo-random number generator (PRNG) or a pseudo-random bit generator (PRBG) is a deterministic algorithm that produces numbers whose distribution is on the one hand indistinguishable from uniform ie. that the probabilities of appearance of the different symbols are equal and that these appearances are all independent. On the other hand, the next output of a PRNG must be unpredictable from all its previous outputs. Indeed, A set of statistical tests for randomness has been proposed in the literature and by NIST to evaluate the security of random(pseudo) bit or block. Unfortunately there are non-random binary streams that pass these standardized tests.

*Corresponding author: E-mail: babacaralassane.ndaw@ucad.edu.sn;
In this paper, as outcome, we introduce on the one hand a new statistical test in a static context called attendance’s law and on the other hand a distinguisher based on this new attendance’s law.

Keywords: Attendance law; Pseudo random number generator; statistical testing.

2010 Mathematics Subject Classification: 94A60; 62F03.

1 Introduction

One family of the cryptographic primitives is random Number Generators (RNG) which have several applications in cryptography such that password generation, nonce generation, Initialisation vector for Stream Cipher, keystream. Recently they are also used to randomise encryption and signature schemes.

A pseudo-random number generator (PRNG) or pseudo-random generator (PRBG) is a deterministic algorithm that produces numbers whose distribution is on the one hand indistinguishable from uniform distribution ie. that the probabilities of appearance of the different symbols are equal and that these appearances are all independent [1, 2, 3, 4, 5, 6, 7]. On the other hand, the next output of a PRNG must be unpredictable from all its previous outputs. To deal with theses properties, there exists two ways to study the PRNG’s security:

1. the randomness of a fixed length of each output called: static security or non-asymptotic security such that NIST test suite [8],
2. the randomness is studying in a dynamic context, the security depends on the size of data. This is called asymptotic security and permits to introduce many new families of PRNG/Hash such that Blum Blum Shub pseudorandom generator family, Keccak(SHA3) [9], FSB [10].

Some studies has been done on both asymptotical and non-asymotical contexts and the relationship between them. In 1982, Yao [11] shows an equivalence between the indistinguishability of a pseudorandom generator and the unpredictability of the next bit from an asymptotic point of view. the static version of this study has been done later in 2009 by S. Ballet and R. Rolland in [12].

The output of any PRNG needs to fit some randomness criteria through statistical tests in order to determine that the PRNG appear random. Theses tests can be applied to a sequence to attemp to compare and evaluate the sequence to a truly random sequence[13, 14]. Indeed, they are in concern of two hypothesis:

- null hypothesis $H_0$ to specify randomness
- alternative hypothesis $H_1$ to specify an non-randomness

For a given test, the answer is to accept or reject null hypothesis. As a reference of statistic test for PRNG and RNG, one can cite the standard A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications by NIST which give criterion and a suite of 15 tests (monobit, block) [8] and [15].

Wang and Nicole [16] pointed out that it is relatively easy to construct some generators whose output is far from uniform distribution, but which passes NIST tests. All these study in the case of indistinguishability where, the number of symbols present on a output stream of a PRNG for a fixed alphabet has not been taken account.

Contribution : In this paper we introduce a new statistic test in a static context based on a probability law called attendance’s law and a distinguisher of PRNG based on it.
Organization: We organize this paper as follows: we start with an introduction Attendance’s law in Section 2. We provide a distinguisher based on it in Section 3. Finally, we conclude in Section 4.

In the following we set $A = \{A_1, A_2, ..., A_n\}$ an alphabet, $\Omega = A^N$ the stream set of length $N$, $\Omega' = \{1, 2, ..., n\}$ hence the number of symbols that appear in a random string $X \in \Omega$ is $1 \leq k \leq n$ and $A^* = \bigcup_{l \in \mathbb{N}} A^l$

2 Related Works: NIST Statistical Test Suite

The NIST statistical test suite is a statistical package \[8\] consisting of 15 tests that were developed to test the randomness of arbitrary long binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. These tests focus on a variety different types of non-randomness that could exist in a sequence. This amounts to checking if the sequence presents statistical biases for certain notions such as bit frequency, entropy, ...

2.0.1 Frequency Test (Monobit or Block):

The sequence is cut into blocks of fixed size $M$. The principle of this test is to calculate the proportion of 1 in these blocks. The expected proportion is $1/2$. Note that if $M = 1$, we fall back on the Frequency Monobit Test.

2.0.2 Runs Test (Monobit or Block):

The principle of this test is to calculate the total number of Runs, that is to say the sequences of the same bit. This amounts to calculating the number of times that 2 consecutive bits are different. Note we take sequence of blocks of fixed size, we fall back on the Runs Block Test.

2.0.3 Binary Matrix Rank Test \[17, 18\]:

The sequence is cut into blocks of size $M$ grouped in groups of $Q$ to form $QxM$ matrices. The rank of these matrices is calculated and the results obtained are compared with a $\chi^2$ reference distribution to measure the linear independence of the blocks.

2.0.4 Template Matching Test \[19, 20\]:

The principle of these tests is to look for the occurrences of the motifs of size $m$. Two different tests are available to us. The sequence is scanned bit by bit until the pattern appears.

For the first, when a pattern is found, we move the window one bit to the right (test patterns overlapping).

For the second, when a pattern is found, we move the window to the first bit following the pattern found (test patterns without overlap). Then, the values obtained are compared to a distribution $\chi^2$. These tests are used to detect generators producing the same patterns in an aperiodic manner.

All these algorithms do not take into account the presence of the alphabet’s symbols in the static stream to be evaluated. In the next section we will introduce a new test that effectively takes this into account.
3 Attendance’s Law

A random generator delivers symbols taken from an alphabet \( A \) of \( n \) symbols (bits, bytes ... etc.) according to a uniform statistical law for example in the uniform distribution the probabilities of appearance of the different symbols are equal to \( \frac{1}{n} \) and that these appearances are all independent.

The law attendance is random variable that allows to evaluate the number of symbols present in a given sequence and distribution [21].

**Definition 3.1.** Let \( Eval : A \times \Omega \rightarrow \mathbb{N} \) such that \((A,X) \mapsto Eval(A,X)\) the number of times "A" appears in \( X \). Let define the law attendance by \( \chi : \Omega \rightarrow \Omega' \) such that \( \chi(X) \) the the number of different symbols in \( X \).

**Proposition 3.1.** Let \( X \in \Omega \) that admits globally \( 1 \leq k \leq n \) fixed different symbols \( \{A_1, A_2, ..., A_k\} \) such that \( Eval(A_i, X) = N_i \) where \( 1 \leq i \leq k \). The number of ways to choose such \( X \) is:

\[
\binom{N}{N_1} \binom{N-N_1}{N_2} \cdots \binom{N-N_1-\cdots-N_{k-1}}{N_k} = \frac{N!}{N_1! N_2! \cdots N_k!}.
\]

and

\[
Pr[X_{i_1} = N_1 \land X_{i_2} = N_2 \land \cdots \land X_{i_k} = N_k | N_1 + N_2 + \cdots + N_k = N] = \frac{N!}{N_1! N_2! \cdots N_k! n^N}.
\]

**Proof.**

On remarks that \( N_1 + N_2 + \cdots + N_k = N \). The number of ways to get \( N_i \) times the the first symbol \( A_i \) from \( X_i \) is \( \binom{N_i}{N_1} \), and \( \binom{N-N_1-\cdots-N_{j-1}}{N_i} \) for the \( j-th \) symbol of \( X_i \) with \( 1 \leq j < k \).

Then the number of ways to obtain \( X_i \) is

\[
\binom{N}{N_1} \binom{N-N_1}{N_2} \cdots \binom{N-N_1-\cdots-N_{k-1}}{N_k} = \frac{N!}{N_1! N_2! \cdots N_k!}.
\]

**Theorem 3.1** (Attendance law). Let \( \chi : \Omega \rightarrow \Omega' \) the attendance law, the probability having globally \( 1 \leq k \leq n \) symbols is:

\[
Pr[\chi = k] = \binom{k}{i} \sum_{i=0}^{k-1} (-1)^i \binom{k}{i} \left( \frac{i}{n} \right)^N
\]

**Proof.** Let \( k \) be an integer s.t. \( 1 \leq k \leq n \)

\[
Pr[\chi = k] = \frac{B_k}{n^N}
\]

where \( B_i \) is the number of sequences of \( i \) symbols from \( A \).

Let \( D_i \) be the number of sequences of a fixed \( i \) symbols, then \( B_i = \binom{\ell}{i} D_i \) and

\[
k^N = \binom{\ell}{1} D_1 + \binom{\ell}{2} D_2 + \cdots + \binom{\ell}{k-1} D_{k-1} + D_k
\]

\[
\Rightarrow D_k = k^N - \binom{\ell}{1} D_1 - \binom{\ell}{2} D_2 - \cdots - \binom{\ell}{k-1} D_{k-1}
\]

As \( D_{k-i} = \frac{B_{k-i}}{\binom{k}{k-i}} \) then

\[
B_k = \binom{k}{1} k^N - \binom{k}{2} \sum_{i=1}^{k-1} \frac{\binom{k-1}{k-1} B_{k-i}}{\binom{k}{i-1}}
\]

\[
= \binom{k}{1} k^N - \sum_{i=1}^{k-1} \frac{\binom{\ell}{k-i} B_{k-i}}{\binom{k}{i-1}}
\]

\[
= \binom{k}{1} k^N - \sum_{i=1}^{k-1} \frac{\binom{\ell}{k-i} B_{k-i}}{\binom{k}{i-1}}
\]
Let \( E_i = \binom{n}{i}^N \), by induction one shows that:

\[
B_k = \sum_{i=1}^{k-1} (-1)^i \binom{n-k+i}{i} E_{k-i}
\]

\[
= \sum_{i=1}^{k-1} (-1)^i \binom{n-k+i}{i} \binom{n}{k-i} (k - i)^N
\]

\[
= \binom{n}{k} \sum_{i=1}^{k-1} (-1)^i \binom{k}{i} (k - i)^N
\]

\[
Pr[\chi = k] = \frac{B_k}{n^N} = \binom{n}{k} \sum_{i=1}^{k-1} (-1)^i \binom{k}{i} \frac{(k - i)^N}{n}
\]

The attendance’s law gives two kinds of distribution depending on parameters \( n \) and \( N \) c.f. figs. 1 and 2.

Fig. 1. Symmetric distribution

Fig. 2. Non-symmetric distribution
4 Attendance Distinguisher

Among the PRNG usage, one of the most frequently used is the construction of stream ciphers. These are symmetric key constructions in which a cipher stream is obtained through the combination of the plaintext bits with a pseudorandom bit sequence. In a stream cipher, the plaintext digits are encrypted one at a time, as opposed to block ciphers, which typically operate on larger chunks of plaintext.

So it becomes necessary to study the random character of the binary sequences generated via different aspects. In this section we will give a formal definition of pseudo-random generators which are obtained from one-way functions and we will evaluate its randomness compared to the law of presence that we defined in the previous section.

Definition 4.1 (Family of One-way Functions). A family of functions \( \{f_m : \mathcal{A}^m \rightarrow \mathcal{A}^N\} \) is One-way with security \( p(m) \) if there is a polynomial time Turing machine that computes them and furthermore for every algorithm \( T \) that runs in time \( p(m) \),

\[
\Pr[T \text{ inverts } f_m(x)] \leq \frac{1}{|\mathcal{A}|} + \frac{1}{p(m)}.
\]

where the probability is taken over uniformly chosen \( \vec{x} \in U(\mathcal{A}^m) \).

Definition 4.2 (Distinguishing a number generator through a single-query). Given a number generator \( g : \mathcal{A}^m \rightarrow \mathcal{A}^N \) with \( N > m \), which expands an \( m \) secret elements random seed into an \( N \) sequences, we define as distinguisher in time \( t \) for \( g \) a probabilistic algorithm \( T \) which, when input with an \( N \) elements string, gives as result either 0 or 1 with time complexity limited by \( t \). We define the advantage of \( T \) for distinguishing \( g \) from a perfect random generator as

\[
\tilde{\text{Adv}}_{\text{prng}}^g(t) = \Pr[\vec{x} \in \mathcal{A}^m (T(g(\vec{x})) = 1)] - \Pr[\vec{y} \in \mathcal{A}^N (T(\vec{y}) = 1)].
\]

The probabilities are considered over the values of a randomly chosen \( \vec{x} \in \mathcal{A}^m \), a randomly chosen \( \vec{y} \in \mathcal{A}^N \), and the random choices of the algorithm \( T \). We state the advantage for distinguishing the function \( g \) in time \( t \) as

\[
\tilde{\text{Adv}}_{\text{prng}}^g(t) = \max_T \{\tilde{\text{Adv}}_{\text{prng}}^g(T)\}.
\]

Definition 4.3 (Pseudorandom Number Generator (PRNG)). A Pseudorandom number generator (PRNG) is a 5-tuple \( <E, V, s_0, f, g> \), where:

- \( E = \mathcal{A}^m \) is a finite state space,
- \( V = \mathcal{A}^N \) is a set of values returned by generator,
- \( s_0 \) is a so-called seed, i.e., initial state in the state sequence \( (s_i)_{i=0}^{\infty} \),
- function \( f : E \rightarrow E \) describes transition between consecutive states \( s_i = f(s_{i-1}) \)
- and \( g : E \rightarrow V \) maps generators state \( s_{i-1} \) into output and \( \tilde{\text{Adv}}_{\text{prng}}^g(t) \) is negligible for \( t \) below a fixed threshold.

Proposition 4.1. Let \( G \) be a PRNG on the alphabet of size \( n \in \mathbb{N} \) and which returns vectors in \( \mathcal{A}^N \) and \( \chi : \Omega \rightarrow \Omega' \) the attendance law on \( \mathcal{A}^N \), then

\[
E[\chi] = \sum_{k=1}^{n} \binom{k}{n} \sum_{i=0}^{k-1} (-1)^i \binom{k-1}{i} \left( \frac{k-1}{n} \right)^N
\]

This proposition gives us for every finite alphabet of size \( n \) and for every generator of size \( N \) on this alphabet the expected average value of the number of characters that must appear in each generator output.
4.1 A Distinguisher of PRNG based on Attendance’s Law

According to the distribution curves of the law of presences, assuming a uniform sequence, the number of presences most likely does not deviate too much from the average, which is, in general, on the side of large values.

4.1.1 Symmetric distribution:

For a symmetric distribution of the law of presences, the probabilities of obtaining the external values are quite weak while that of getting the central value of $k$ is inspecting to be very large.

One can then set a probability $\alpha$, which is call the level of significance [8] or degree of confidence, such as the probability that the number of presences $k$ is outside the region of great values (region acceptance of the hypothesis $H_0$) is less than or equal to $\alpha$.

4.1.2 Non-symmetric distribution:

For a non-symmetric distribution, the average being on the side of the big extreme values of $k$, One can therefore, set a significance threshold $\alpha$ such as the probability that the number of presences $k$ is in the region of low values either less than or equal to $\alpha$.

Definition 4.4. Let $\epsilon$ be a positive number, $\chi$ the attendance law. We define the acceptance area relatively to $\epsilon$:

$$A_\epsilon = \{ X \in \Omega : 0 \leq k_0 - \epsilon \leq X \leq k_0 + \epsilon \leq n \}$$

where $k_0 = \lfloor E[\chi] \rfloor$

As noted above, the Distinguisher $D$ which allows to rule on the uniformity or not of the output keystream of a generator, is a hypothesis test which, for a keystream sample given $X = X_1X_2...X_N$ allows to decide whether to accept or reject $H_0$.

**Inputs:** $X = X_1X_2...X_N \in \Omega$

**Outputs:** \{SUCCESS, FAILURE\}

**Parameters:** $n, N \in \mathbb{N}, \alpha \in [0,1]$

1. Compute $k = \chi(X)$
2. Compute $P_{value} = Pr[\chi = k]$
3. Compare the $P_{value}$ to $\alpha$:
   - return SUCCESS whenever $P_{value} > \alpha$
   - return FAILURE whenever $P_{value} < \alpha$

![Fig. 3. Statistical Test on Attendance’s Law](image)

5 Experimental Results

In this section we give some experimental results based on our distinguisher and on another taken from the NIST SP800-22 suite based on the same fixed bitstream for a fair comparison. We also give an evaluation of Linear congruential generators (LCG) based on the Microsoft LCG C/C++ implementation.
Table 1. Some Acceptance areas

| N   | 𝑛   | 95%      | 99%      |
|------|------|----------|----------|
| 25   | 26   | 14-26    | 13-26    |
| 50   | 26   | 20-26    | 19-26    |
| 100  | 26   | 23-26    | 22-26    |
| 50   | 27   | 19-27    | 18-27    |
| 100  | 31   | 27-31    | 26-31    |
| 100  | 32   | 28-32    | 27-32    |
| 200  | 26   | 25-26    | 25-26    |
| 1250 | 256  | 251-256  | 250-256  |
| 2500 | 256  | 255-256  | 255-256  |

5.1 NIST Frequency Test Within Block Vs Our Attendance Test

\[\epsilon = 1100100100111101110110101000100010110100011000100010001000101101000110001000110100110001001100011001100010100010111000\]

for \(\alpha = 0.01\):

Table 2. Comparison with NIST Frequency Test

| \(\epsilon\) | NIST Frequency Test | Our Attendance Test \((n = 2^4)\) | Our Attendance Test \((n = 2^4)\) | Our Attendance Test \((n = 2^4)\) |
|-------------|---------------------|----------------------------------|----------------------------------|----------------------------------|
| accepted    | accepted            | Rejected                         | Rejected                         |

It follows from the above that this test is independent of those of NIST that it is therefore quite possible that a binary suite passes the NIST tests and is rejected by the law of attendance.

5.2 Linear congruential generators (LCG)

As an application example, we propose to evaluate LCG generator.

Linear congruential generators (LCG) update their state according to the following recursive formula

\[s_n = (as_{n-1} + c) \mod M\]

Generators from this class are defined by three integer parameters a modulus \(M\), a multiplier \(a\) and an additive constant \(c\) and are shortly denoted as \(LCG(M, a, c)\). The method represents one of the oldest and best-known pseudorandom number generator algorithms.

Two historic random number generators has been replicated. One is the \(rand()\) function from BSD libc, and the other is the \(rand()\) function from the Microsoft C Runtime (MSCVRT.DLL). Each replica give the same sequence of integers as the original generator, when starting from the same seed.
5.2.1 Microsoft LCG C/C++ implementation

For $LCG(2^{32}, 214013, 2531011)$ we get the table 3:

$$
\epsilon = 01011111001001111001101100001111101101101100100100110101111001101110110110111010110101010000001000111100111010101001111100110110
$$

for $\alpha = 0.01$:

| N | n | k | $P$-value | result     |
|---|---|---|-----------|------------|
| 100 | 2 | 2 | 1         | random     |
| 100 | $2^4$ | 4 | 0.99      | random     |
| 100 | $2^8$ | 8 | 0.99      | random     |
| 100 | $2^{14}$ | 14 | 0.97     | not random |
| 100 | $2^{17}$ | 17 | 0.23     | not random |

According to our Distinguisher and the table 3, the sampling in Microsoft LCD version is random for $n = 2, 4, 8$ (monobit, 2-bit alphabet or 3-bit alphabet) and for $n > 8$ the sampling is not done correctly.

6 Conclusion

In this paper we proposed a novel method of testing PRNGs which is based on the attendance’s law and establish a distinguisher. Unlike the other statistical tests which deals on the sequential aspect of the streams, with the law of attendance we have given properties on the way in which symbols are chosen.

The experimental results presented in this paper show that our testing procedure can be used for detecting weaknesses in many Random number generator depending on the chosen alphabet.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Shafi Goldwasser, Silvio Micali. Probabilistic encryption. Journal of Computer and System Sciences. 1984;28(2):270299.
[2] Oded Goldreich. Modern cryptography, probabilistic proofs and pseudorandomness. Springer-Verlag, Berlin, Heidelberg; 1998.
[3] Oded Goldreich. A note on computational indistinguishability. Information Processing Letters. 1990;34(6):277281.
[4] Goldreich Oded. Foundations of cryptography: Volume 2, basic applications. Cambridge University Press, New York, NY, USA; 2009.
[5] Oded Goldreich. Foundations of cryptography: Volume 1. Cambridge University Press, New York, NY, USA; 2006.
[6] Rolf Oppliger. Contemporary cryptography. Artech House computer security series. Artech House; 2005.
[7] Oded Goldreich. On the foundations of modern cryptography. In Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, CRYPTO 97. London, UK, UK. 1997:4674. Springer-Verlag.

[8] Lawrence E. Bassham, Andrew L. Rukhin, Juan Soto, James R. Nechvatal, Miles E. Smid, Elaine B. Barker, Stefan D. Leigh, Mark Levenson, Mark Vangel, David L. Banks, Nathanael Alan Heckert, James F. Dray, San Vo. Sp 800-22 rev. 1a, a statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical Report, Gaithersburg, MD, USA; 2010.

[9] Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, Editors. Advances in Cryptology (EUROCRYPT) Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 2013;313314.

[10] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A fast provably secure cryptographic hash function. IACR Cryptology ePrint Archive, 2003:230, 2003.

[11] Andrew Chi-Chih Yao. Theory and application of trapdoor functions. In 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). 1982:8091.

[12] Stephane Ballet, Robert Rolland. A note on yaos theorem about pseudo-random generators. Cryptography and Communications. 2011;3(4):189.

[13] Renaud Santoro. Towards high-rate uniform and gaussian random number generators. Theses. Universite Rennes; 2009.

[14] Patrick Lacharme. Ghrateur vraiment alatoire dans un composant scuris. PhD thesis, 2007. Thse de doctorat dirige par Langevin, Philippe Informatique Toulon; 2007.

[15] John Michael Kelsey Elaine B Barker. Nist special publication 800-90a , rev . 1 - recommendation for random number generation using deterministic random bit generators. Tech. rep., NIST; 2015.

[16] Yongge Wang, Tony Nicol. On statistical distance based testing of pseudo random sequences and experiments with php and debian openssl. Computers & Security. 2015;53:4464.

[17] Igor Kovalenko. Distribution of the linear rank of a random matrix. Theory of Probability & Its Applications. 1973;17(2):342346.

[18] George Marsaglia, Liang-Huei Tsay. Matrices and the structure of random number sequences. Linear Algebra and its Applications. 1985;67:147156.

[19] Ourania Chryssaphinou, Stavros Papastavridis. A limit theorem on the number of overlapping appearances of a pattern in a sequence of independent trials. Probability Theory and Related Fields. 1988;79:129 143.

[20] Kenji Hamano, Toshinobu Kaneko. Correction of overlapping template matching test included in nist randomness test suite. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 2007;E90-A(9):17881792.

[21] Babacar Alasane Ndaw. Generateurs d’aleaet Securite: Loi de Presences - Conception d’un distinguier d’alea. PhD thesis; 2016.