RW-9: A Family of Random Walk Tests

Muhidin Uğuz
Middle East Technical University
Ankara, TURKEY
muhid@metu.edu.tr

Fatih Sulak
Atilim University
Ankara, TURKEY
fatih.sulak@atilim.edu.tr

Ali Doğanaksoy
Middle East Technical University
Ankara, TURKEY
aldoks@metu.edu.tr

Onur Koçak
TÜBİTAK
Ankara, TURKEY
onur.kocak@tubitak.gov.tr

Abstract

In this work, we define a family of 9 statistical randomness tests for collections of short binary strings, by making use of random walk statistics. For a binary sequence of length n we consider the probability of intersecting the line $y = t$ exactly at k distinct points. In the literature there are some explicit formulas for these probability values but the ones for short sequences are not feasible for computations concerning sequences of length 256 or more. On the other hand, approximation techniques, or asymptotic approaches that should be used only when testing long sequences, are not useful for testing sequences of length between 256 and 4096. Recursive formulas, derived in this paper, made it possible to obtain exact values of the corresponding probability distribution functions. Employing these formulas, we have provided necessary figures for testing collections of strings of length 2^7, 2^8, 2^{10} and 2^{12} bits. Finally we have applied these 9 tests to several collections of strings obtained from different pseudorandom number generators and to biased sequences in order to see if the tests introduced can detect non-random data.

Keywords: Cryptography, Random Walk, Statistical Randomness Testing, NIST Test Suite

1 Introduction

The quality of a binary sequence, produced by a pseudorandom number generator to be used as a seed for cryptosystems, has a vital importance. It should be random looking, that is, should not follow any pattern that may give rise to an attack to the system. Moreover, outputs of encryption algorithms must also be indistinguishable from a true random sequence. Thus, in evaluation of a binary sequence, a pseudorandom number generator or an encryption algorithm in terms of randomness of its output, one needs a statistical randomness test or even a package of tests that evaluates the randomness property. Since any periodic sequence may appear as an output of a true random source, there is no mathematical method to decide whether the sequence under consideration is in fact an output of a true random generator or not, for sure. On the other hand, one can give decision depending on the observed statistical values of the sequence, comparing it with the expected values and distributions. There are many statistical randomness tests and test suites in the literature ([1], [2], [3]). Moreover there are many studies about the independence of the statistical randomness tests ([4], [5]). Different point of views yields different statistical tests. One of these point of views is random walk statistics. In the literature there are various randomness tests concerning random walk statistics, however these tests can either be applied to a very long sequence (a sequence of length at least 10^6), such as the tests included in the NIST Test Suite [6], or to a collection of very short sequences (sequences of length at most 256), as stated in [7]. Random walk tests are included in the NIST Test Suite in the name of Random Excursion and Random Excursion Variant Tests [6]. These tests use approximations in the computations of the cumulative probability distribution of the corresponding random variables and therefore can be applied only to sequences of length longer than 10^6. In this work, we revisit the distribution function of the test and give the exact probability values for sequences of size less then 4096.

The organization of the paper is as follows. In section 2, we give preliminaries. The recursive relation satisfied by the number of sequences of length n, having exactly k balanced points is derived first for balanced sequences in section 3, and then for general sequence in section 4, to be used for the random excursion statistics. In section 5, using the results obtained in sections 3 and 4, we derive recursive relations for the number of strings intersecting the line $y = t$, exactly k times. In section 6, we define new randomness tests based on these statistics. In section 7, we apply the proposed tests to random and non-random data. Finally in section 8, we conclude the paper. We omitted the proofs of Lemma 4, Theorem 11, Proposition 12 and Theorem 13 due to the page limitations.

2 Preliminaries

First, we will introduce the notations used in this article. σ denotes a binary string $s_1, s_2, \cdots, s_n$ of length n. A string σ is called *balanced* if the number of its terms, that is s_i's, equal to 0 is the same as the number of its terms equal to 1; we call a term s_k $(k \leq n)$ a *balance point* if the substring $s_1, \ldots, s_k$ is balanced. Obviously a string is balanced if and only if the last term s_n is a balance point.

To give a motivation for the following definitions, consider a binary string $\sigma = s_1, \cdots, s_n$ of length n, and its graph. The graph of a sequence σ, regarded as a continuous function, is drawn by joining the gaps between dots (i, s_i) with line segments. These line segments start from the origin, and the part of it between $(i-1, s_{i-1})$ and (i, s_i) has slope equal to $a_i = (-1)^{s_i} = 1 - 2s_i$. The graph of the sequence $\sigma = 0, 1, 1, 0, 0, 1, 0, 0$ is given below to illustrate this method of drawing graph of a discrete binary sequence, regarding

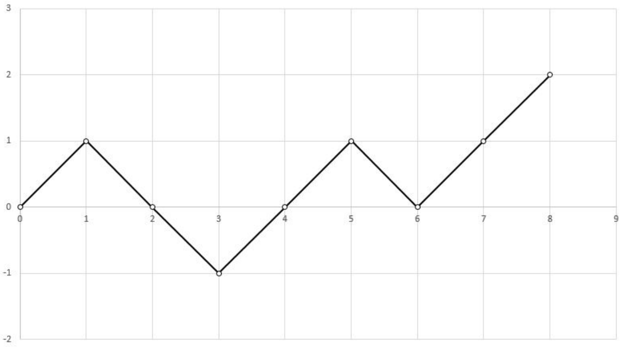

Figure 1: Graph of the sequence $\sigma = 0, 1, 1, 0, 0, 0, 1, 0, 0$

it as a continuous function.

Definition 1. A string σ is said to **intersect** (meet or touch) the line $y = t$ at $x = i$ if $2(s_1 + \cdots + s_i) = i - t$, that is the difference between the ones and zeroes in the ordered set $\{s_1, s_2, \ldots, s_i\}$ is t.

Note that s_i is a *balance point* of the binary sequence σ if and only if the graph of σ intersects the line $y = 0$ at $x = i$.

Definition 2. Here we list some definitions and notations that will be used throughout the paper.

- $\mathbf{X_t(n, k)}$ denotes the set of all strings of length n which intersect the line $y = t$ exactly at k distinct terms and denote the number of elements of this set, that is $|X_t(n, k)|$, by $x_t(n, k)$. As a special case, for $k = 0$, we write $X_t(n) = X_t(n, 0)$ and similarly, $x_t(n) = |X_t(n, 0)|$. Since $X_t(n)$ is the set of strings of length n which do not intersect the line $y = t$, the complement $\overline{X}_t(n)$ of this set consists of strings which intersect the line $y = t$ at least in one point. We write $\overline{x}_t(n) = |\overline{X}_t(n)|$.

- $\mathbf{B(n, k)} \subset X_0(n, k)$ denotes the set of balanced strings of length n which contain exactly k balance points and $b(n, k)$ is the number of such strings.

- $\mathbf{B_t(n)}$ stands for the set of strings of length n which touch the line $y = t$ for the first time at the last term and $b_t(n) = |B_t(n)|$. Note that $B_0(n) = B(n, 1)$ and if $t \neq 0$, then no string in $B_t(n)$ is balanced. In fact, if $\sigma \in B_t(n)$, then $\dfrac{n+t}{2}$ of the terms of σ are equal to zero.

- $\mathbf{X(n, k)}$ denotes the set of strings which contain exactly k balance points and $x(n, k)$ is the number of such strings.

- The probability of a string of length n to have exactly k intersections with the line $y = t$ (or $y = -t$) will be denoted by $\mathbf{p_t(n, k)} = prob(\sigma \in X_t(n, k)) = x_t(n, k)/2^n$.

- Let n and k be positive integers and let $a(i,j)$ be a two dimensional array, $i = 1,\dots,n$ and $j = 1,\dots,k$. By $[\mathbf{a}(\mathbf{n},\mathbf{k})]$ we denote the table (matrix) whose rows are indexed by $i = 1,\dots,n$ and columns are indexed by $j = 1,\dots,k$. In certain circumstances row and/or column indices are allowed to start with 0 rather than 1.

From the definition it follows that $x_0(n,k) = x(n,k)$ and $x_t(n,k) = x_{-t}(n,k)$ for any $t = 1,\dots,n$.

One of the basic tools we are going to employ is the sequence $\{C_n\}_{n=0}^{\infty}$of Catalan numbers defined by $C_n = \dfrac{1}{n+1}\dbinom{2n}{n}$ for any non negative integer n. The first few terms of this sequence are 1, 1, 2, 5, 14,….

Now we will summarize some well known identities about Catalan numbers. It is straightforward to see that the Catalan numbers satisfy the following recursion:

$$C_n = \frac{4n-2}{(n+1)}C_{n-1}\ \forall n > 1. \tag{1}$$

Another important property of the Catalan numbers is that, convolution of the sequence $\{C_n\}_{n=0}^{\infty}$ with itself is again itself, that is, for any non negative integer n, $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$. Moreover, using this convolution property, it is east to show that the generating function of this sequence,

$$C(z) = \sum_{i=0}^{\infty} C_i z^i = 1 + z + 2z^2 + 5z^3 + 14z^4 + \cdots$$

satisfies the equation

$$zC^2(z) = C(z) - 1 \tag{2}$$

and from which it follows that $C(z) = \frac{1-\sqrt{1-4z}}{2z}$.

By differentiating both sides of the equation (2) one obtains

$$\frac{d}{dz}C(z) = \frac{C^2(z)}{1-2zC(z)} = \frac{C(z)-1}{z(1-2zC(z))} \tag{3}$$

and by differentiating the product

$$zC(z) = \sum_{i=0}^{\infty} \frac{1}{i+1}\binom{2i}{i} z^{i+1}$$

we obtain the generating function of the sequence $\left\{\binom{2n}{n}\right\}_{n=0}^{\infty}$ as:

$$C(z) + z\frac{d}{dz}C(z) = \sum_{i=0}^{\infty} \binom{2i}{i} z^i. \tag{4}$$

The following lemma presents a result which will be the basis of many computations throughout the work.

Lemma 3. *Let n, t and q be positive integers with $t \leq q \leq n$. The number of strings of length n which contain q zeros and which intersect the line $y = t$ at least once is given by*

$$\begin{cases} \dbinom{n}{q-t} & \textit{if } q < \dfrac{n+t}{2}, \\[2ex] \dbinom{n}{q} & \textit{if } \frac{n+t}{2} \leq q \leq n. \end{cases}$$

Proof. Given a string σ of length n which intersects the line $y = t$, depending on q we consider two cases:

1. $\dfrac{n+t}{2} < q \leq n$. In this case σ necessarily intersects the line $y = t$ and number of such strings is $\dbinom{n}{q}$.

2. $t \leq q \leq \dfrac{n+t}{2}$. Let A be the set of strings of length n which have q zeros and which intersect the line $y = t$, and let B be the set of strings of length n which have $q - t$ zeros. We will show that these two sets have the same number of elements, so that the number of strings in A is $\dbinom{n}{q-t}$.

   Given $\sigma \in A$. Let i_0 be the smallest integer such that σ intersects the line $y = t$ at s_{i_0}. The string $\overline{\sigma} = \overline{s}_1 \cdots \overline{s}_{i_0} s_{i_0+1} \cdots s_n$ where $\overline{s}_i = 1 - s_i, i = 1, \ldots, i_0$ has $q - t$ zeros, hence $\sigma \in B$. Thus, to each $\sigma \in A$, there corresponds a unique string $\overline{\sigma} \in B$. Conversely, any string τ in B has $q - t$ zeros, hence $n - q + t$ ones. On the other hand, the condition $q \leq (n + t)/2$ implies that $n - q + t \geq (n + t)/2$, which means that the string τ intersects the line $y = -t$. Now in the string τ, starting with the first term replace each one with a zero and each zero with a one up to the term at which the string intersects the line $y = -t$ for the first time. The resulting string intersects the line $y = t$ and has q zeros, hence is in A. Then the correspondence given above is one to one and the sets A and B have the same number of elements.

□

Lemma 4. *Let n and t be positive integers with $t \leq n$. The number of strings of length n which intersect the line $y = t$ at least once is given by*

$$\overline{x}_t(n) = \begin{cases} 2\displaystyle\sum_{i=0}^{\frac{n-t}{2}} \binom{n}{i} - \binom{n}{\frac{n-t}{2}} & \textit{if } n+t \textit{ is even}, \\ 2\displaystyle\sum_{i=0}^{\frac{n-t-1}{2}} \binom{n}{i} & \textit{if } n+t \textit{ is odd}. \end{cases} \tag{5}$$

3 Recursive Relations Satisfied by $b(n, k)$

We first give an explicit expression for $b(n, 1)$, the number of balanced sequences which have no balance points other than the last term. It is obvious that a balanced sequence must be of even length, therefore $b(n, 1) = 0$ for any odd integer n. For sequences of even length we have the following proposition.

Proposition 5. *For any positive integer m, $b(2m, 1) = 2C_{m-1}$ where C_{m-1} denotes the corresponding Catalan number.*

Proof. Any $\sigma = s_1 \cdots s_{2m} \in B(2m, 1)$ is balanced and has only one balance point (necessarily the last term) and none of the terms $s_1, \ldots, s_{2m-1}$ is a balance point. For $m = 1$ the claim is apparent: $b(2, 1) = 2 = 2C_0$. Now assume that $m > 1$ and $s_1 = 1$ (hence, $s_{2m} = 0$). It easy to see that the string $s_2, \ldots, s_{2m-1}$ is balanced and it cannot intersect the line $y = 1$. Thus, to each such string, there corresponds a unique string $\sigma \in B(2m, 1)$ with $s_1 = 1$. Since the converse relation holds also, the number of strings in $B(2m, 1)$ is equal to the number of strings of length $2m - 2$ which have $q = m - 1$ zeros and which do not intersect the line $y = 1$. Then, from Lemma 3 we obtain

$$b(2m, 1) = \binom{2m-2}{m-1} - \binom{2m-2}{m-2}$$

which simplifies into C_{m-1}. By including the strings with initial term 0, the assertion follows. □

As a result, for any nonnegative integer we have

$$b(n, 1) = \begin{cases} 0 & \textit{if } n = 0 \textit{ or } n \textit{ is odd}, \\ 2C_{\frac{n}{2}-1} & \textit{if } n > 0 \textit{ is even}. \end{cases} \tag{6}$$

Proposition 6. *For any positive integers m and k >1, the sequence $\{b(2m, k)\}_{n=0}^{\infty}$ is the convolution of the sequences $\{b(n, 1)\}_{n=0}^{\infty}$ and $\{b(n, k-1)\}_{n=0}^{\infty}$, that is*

$$b(2m, k) = \sum_{i=0}^{m-1} b(2i, 1) b(2m - 2i, k - 1).$$

Proof. Let $k > 1$ and consider a string $\sigma \in B(n, k)$. Assume that the first balance point is s_{2i}. Then, σ can be separated into two substrings $\sigma_1 = s_1 \cdots s_{2i}$ and $\sigma_2 = s_{2i+1} \cdots s_{2m}$ such that $\sigma_1 \in B(2i, 1)$ and $\sigma_2 \in B(2m - 2i, k - 1)$. □

Now, we focus on the generating function of the sequence $\{b(n, k)\}_{n=0}^{\infty}$. First we find the generating function $B(z)$ of $\{b(n, 1)\}_{n=0}^{\infty}$:

$$\begin{aligned} B(z) &= \sum_{i=0}^{\infty} b(i, 1) z^i = \sum_{i=1}^{\infty} b(2i, 1) z^{2i} = 2 \sum_{i=1}^{\infty} C_{i-1} z^{2i} = 2z^2 \sum_{i=0}^{\infty} C_i z^{2i} = 2z^2 C(z^2) \\ &= 1 - \sqrt{1 - 4z^2}. \end{aligned}$$

Proposition 7. *Let k be a positive integer. Then generating function of the sequence $\{b(n,k)\}_{n=0}^{\infty}$ is $B^k(z) = 2^k z^{2k} C^k(z^2)$.*

Proof. Proposition 6 implies that the generating function of $\{b(n,k)\}_{n=0}^{\infty}$ is the product of $B(z)$ and the generating function of $\{b(n,k-1)\}_{n=0}^{\infty}$. Then,the proof follows inductively: generating function of $\{b(n,k)\}_{n=0}^{\infty}$ for $k=2$ is $B(z)B(z) = B^2(z)$. For $k=3$ we have $B(z)B^2(z) = B^3(z)$ and so on. □

For the sake of completeness we let $B^0(z)$ be the identity function.

Theorem 8. *For any positive integers n and k, the quantities $b(n,k)$ satisfy the following recursions subject to the given initial conditions.*

1. *For $k=1$*

$$b(n,1) = \begin{cases} 0 & if\ n=1, \\ 2 & if\ n=2, \\ \dfrac{4(n-3)}{n} b(n-2,1) & if\ n \geq 3. \end{cases}$$

2. *For $k=2$*

$$b(n,2) = \begin{cases} 0 & if\ n \leq 2, \\ 2b(n,1) & if\ n \geq 3. \end{cases}$$

3. *For $k \geq 3$*

$$b(n,k) = \begin{cases} 0 & if\ n < 2k, \\ 2b(n,k-1) - 4b(n-2,k-2) & if\ n \geq 2k. \end{cases}$$

Proof. 1. Initial terms are obvious and the recursion follows from (1) and (4).

2. Initial terms are obvious. Generating function of $\{b(n,2)\}_n$ satisfies

$$\begin{aligned} B^2(z) &= 4x^4C^2(z^2) = 4z^2\left(z^2C^2(z^2)\right) = 4z^2\left(C(z^2)-1\right) = 4z^2C(z^2) - 4z^2 \\ &= 2B(z) - 4z^2 \end{aligned}$$

which means that $b(2,2) = 2b(2,1) - 4 = 0$ and for $n > 2$, $b(n,2) = 2b(n,1)$.

3. Initial terms are obvious. For any integer $k > 2$ we have

$$\begin{aligned} B^k(z) &= 2^k z^{2k} C^k(z^2) = 2^k z^{2k-2}[C^{k-2}(z^2)][z^2C^2(z^2)] = 2^k z^{2k-2}[C^{k-2}(z^2)][C(z^2)-1] \\ &= 2\left(2^{k-1}z^{2k-2}C^{k-1}(z^2)\right) - 4z^2\left(2^{k-2}z^{2k-4}C^{k-2}(z^2)\right) \\ &= 2B^{k-1}(z) - 4z^2B^{k-2}(z) \end{aligned}$$

which implies that $b(n,k) = 2b(n,k-1) - 4b(n-2,k-2)$ for any integer $n > 2$. □

4 Recursive Relations Satisfied by $x(n, k)$

Given a positive integer n, by substituting $t = 1$ in Equation (5) we observe that

$$\overline{x}_1(n) = \begin{cases} 2^n - \binom{n}{\frac{n-1}{2}} & if\ n\ is\ odd \\ 2^n - \binom{n}{\frac{n}{2}} & if\ n\ is\ even \end{cases}$$

which can be written simply as $\overline{x}_1(n) = 2^n - \binom{n}{\lfloor \frac{n}{2} \rfloor}$. On the other hand, by definition, $x_1(n) = 2^n - \overline{x}_t(n)$ which gives the number of strings which do not intersect the line $y = 1$ as

$$x_1(n, 0) = x_1(n) = \binom{n}{\lfloor \frac{n}{2} \rfloor}. \tag{7}$$

Now, let $\sigma \in X_0(n)$ and assume that $s_1 = 1$, then $s_2 \dots s_n \in X_1(n-1, 0)$. It follows that the number of strings in $X_0(n)$ with the first term 0 is $x_1(n-1, 0) = \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$. Since the same holds for the strings with the first term 1, we obtain the number of strings which do not intersect the line $y = 0$ as

$$x_0(n, 0) = x_0(n) = 2\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}. \tag{8}$$

Let $X_k(z)$ be the generating function of the sequence $\{x(n, k)\}_{n=0}^{\infty}$ and for the special case $k = 0$ write $X(z) = X_0(z)$. We have $X(z) = \sum_{i=0}^{\infty} x(i, 0) z^i$, where we let $x(0, 0) = 1$. We can write this function as $X(z) = \sum_{i=0}^{\infty} x(2i, 0) z^{2i} + x(2i + 1, 0) z^{2i+1}$. From (5) we obtain $x(2i, 0) = 2\binom{2i-1}{i} = \binom{2i}{i}$ and $x(2i + 1, 0) = 2\binom{2i}{i}$, thus

$$X(z) = \sum_{i=0}^{\infty} \left(\binom{2i}{i} + 2\binom{2i}{i} z \right) z^{2n} = (1 + 2z) \sum_{i=0}^{\infty} \binom{2i}{i} z^{2i}.$$

Now, from (3) we write $\sum_{i=1}^{\infty} \binom{2i}{i} z^{2i} = C(z^2) + z^2 C'(z^2)$ which leads to

$$X(z) = (1 + 2z)(C(z^2) + z^2 C'(z^2)). \tag{9}$$

Using the substitutions $z^2 C'(2) = \frac{C(z^2)-1}{1-2z^2C(z^2)}$ and $z^2 C^2(z^2) = C(z^2) - 1$ in (8):

$$\begin{aligned} X(z) &= (1 + 2z) \left(C(z^2) + \frac{C(z^2) - 1}{1 - 2z^2 C(z^2)} \right) = (1 + 2z) \left(\frac{2C(z^2) - 2z^2 C^2(z^2) - 1}{1 - 2z^2 C(z^2)} \right) \\ &= \frac{1 + 2z}{1 - 2z^2 C(z^2)} = \frac{1 + 2z}{1 - B(z)} = \frac{1 + 2z}{\sqrt{1 - 4z^2}} = \sqrt{\left(\frac{1 + 2z}{1 - 2z} \right)} \end{aligned}$$

Proposition 9. *For any positive integer n, $x(n, 0) = 2\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}$, and for any integer k >1, $x(n, k) = \sum_{i=1}^{\lfloor n/2 \rfloor} b(2i, k - 1) x(n - 2i, 0)$.*

Proof. Let $k > 1$ and consider a string $\sigma \in X(n,k)$. Assume that the last balance point is s_{2i}. Then, σ can be separated into two substrings $\sigma_1 = s_1 \cdots s_{2i}$ and $\sigma_2 = s_{2i+1} \cdots s_n$ such that $\sigma_1 \in B(2i,k)$ and $\sigma_2 \in B(n-2i,k-1)$.

□

Proposition 10. *For any positive integer k, generating function of the sequence $\{x(n,k)\}_{n=0}^{\infty}$ is $X_k(z) = X(z)B^k(z)$.*

Proof. Previous proposition implies that $X_k(z) = X_{k-1}(z)B(z)$. Then, for $X_1(z) = X(z)B(z)$ and the assertion follows inductively. □

With the notation of the above proposition, if we substitute $k = 0$, we see that $X_0(z) = X(z)B^0(z) = X(z)$.

Theorem 11. *For any nonnegative integers n and k, the quantities $x(n,k)$ satisfy the following recursions subject to the given initial conditions:*

$$k = 0 \implies x(n,0) = \begin{cases} 1 & if\ n = 0 \\ 2 & if\ n = 1 \\ 2(1-\dfrac{1}{n})x(n-1,0) & if\ n \geq 2\ is\ even \\ 2x(n-1,0) & if\ n \geq 3\ is\ odd \end{cases}$$

$$k = 1 \implies x(n,1) = \begin{cases} 0 & if\ n \leq 1 \\ x(n,0) & if\ n \geq 2 \end{cases}$$

$$k \geq 2 \implies x(n,k) = \begin{cases} 0 & if\ n < 2k \\ 2x(n,k-1) - 4x(n-2,k-2) & if\ n \geq 2k \end{cases}$$

5 Recursive Relations Satisfied by $x_t(n,k)$

We have defined $X_t(n,k)$ to be the set of strings which intersect the line $y = t$ at exactly k terms. For $t = 0$ we have already obtained recursive relations by which $[x_0(n,k)]$ can be computed effectively. So, we focus on the case $t \neq 0$ and since $x_{-t}(n,k) = x_t(n,k)$, without loss of generality we can assume that t is positive.

Proposition 12. *Given integers $n, k \geq 0$ and $t > 0$. If $n < t + 2k - 2$, then $x_t(n,k) = 0$. If $n \geq t + 2k - 2$, then $x_1(n,k) = \frac{1}{2}x(n+1,k)$,*

$$x_2(n,k) = \begin{cases} x_1(n+1,0) & if\ k = 0 \\ x_1(n+1,k) - x(n,k-1) & if\ k \geq 1 \end{cases}$$

$$x_t(n,k) = x_{t-1}(n+1,k) - x_{t-2}(n,k) \qquad (t \geq 3)$$

Theorem 13. *Let $n \geq 0, k \geq 0$ and $t \geq 1$ be integers. The numbers $x_t(n,k)$ satisfy the following recursions:*

$$x_t(n,k) = \begin{cases} x_1(n,k) = \dfrac{1}{2}x(n+1,k) & \mathit{if}\ t = 1 \\ \dfrac{1}{2}(x_{t-1}(n+1,0) + x_{t-1}(n+1,1)) & \mathit{if}\ t \geq 2\ \mathit{and}\ k = 0 \\ x_t(n,k) = \dfrac{1}{2}x_{t-1}(n+1,k+1) & \mathit{if}\ t \geq 2\ \mathit{and}\ k \geq 1 \end{cases}$$

Theorem 14. *Let n, k, and t be nonnegative integers. The table $[p_t(n,k)]$ can be constructed by the following recursions*

i. $t = 0$ *and* $k = 0 \implies p_0(n,0) = \begin{cases} 1 & \mathit{if}\ n = 0, \\ 1 & \mathit{if}\ n = 1, \\ (1 - \frac{1}{n})x_0(n-1,0) & \mathit{if}\ n \geq 2\ \mathit{is\ even}, \\ p_0(n-1,0) & \mathit{if}\ n \geq 3\ \mathit{is\ odd}. \end{cases}$

ii. $t = 0$ *and* $k = 1 \implies p_0(n,1) = \begin{cases} 0 & \mathit{if}\ n \leq 1, \\ p_0x(n,0) & \mathit{if}\ n \geq 2. \end{cases}$

iii. $t = 0$ *and* $k \geq 2 \implies p_0(n,1) = \begin{cases} 0 & \mathit{if}\ n < 2k, \\ 2p_0(n,k-1) - p_0(n-2,k-2) & \mathit{if}\ n \geq 2k. \end{cases}$

iv. $t = 1 \implies p_1(n,k) = p_0(n+1,k)$.

v. $t \geq 2$ *and* $k = 1 \implies p_t(n,0) = p_{t-1}(n+1,0) + p_{t-1}(n,1)$.

vi. $t \geq 2$ *and* $k \geq 2 \implies p_t(n,k) = p_{t-1}(n+1,k+1)$.

Proof. Just substitute $p_t(n,k) = 2^{-n}x_t(n,k)$ in Theorem 11 and Theorem 13. □

6 RW-9 Random Walk Tests

We propose a family of randomness tests based on random walk statistics, namely RW-9 random walk tests. RW-9 random walk tests first convert a binary string to a random walk and then count the number of times that a random walk intersect the line $y = t$. The input of the test is a collection of binary strings with equal length n. We apply the test function to determine the number of intersections with the line $y = t$ in each string, and call them as observed values. Afterwards, we apply χ^2 test and produce p-value using the bin probability tables (as described in [8]). We give the probabilities for $n \in \{128, 256, 1024, 4096\}$. It should be noted that the 9 test statistics defined in this paper are not necessarily independent.

6.1 Walkthrough

Tables 2, 3, 4, and 5 present the number of bins, bin values and the probabilities corresponding to each bins, for $n = 128, 256, 1024$ and 4096 respectively. As an example, to test the randomness of a collection of N binary strings of length $n = 128$, the first row of Table 2, that is the line labeled as "$y = 0$" suggests the use of 8 bins, and gives the expected values of the number of excursions to be 0 or 1 as $0.140772 \times N$, to be 2 or 3 as $0.138555 \times N$, ..., to be between 17 and 128 as $0.107782 \times N$.

The procedure to test a collection of N binary strings of length n, using the Random Walk Tests family can be summarized as follows:

1. Determine the corresponding number of bins for each of the test functions, that is, the number of intersections of the random walk with the line $y \in \{0, \pm1, \pm2, \pm3, \pm4\}$.

2. Apply χ^2 Goodness of Fit Test, that is evaluate

$$\chi^2 = \sum_{i=1}^{B} \frac{(O_i - N \cdot p_i)^2}{N \cdot p_i} \qquad \text{and} \qquad p\text{-value} = \mathtt{igamc}\left(\frac{B-1}{2}, \frac{\chi^2}{2}\right)$$

   where p_i's are obtained from bin probability tables 2, 3, 4, and 5.

3. If p-value$< \epsilon$, conclude that the null hypothesis H_0 (the randomness hypothesis) is rejected, otherwise accepted. In cryptographic applications, ϵ is usually set to 0.01.

7 Application

This section reveals the results obtained from the application of the Random Walk Tests to various collections of strings in order to show the sensitivity of the tests. For this purpose, we generate pseudorandom and non-random data sets. The details are as follows.

First, we apply the tests on the outputs of AES-128, SHA-2 Family and MD5 which are considered as random looking. For generating AES-128 outputs, 128-bit representations of the numbers from 0 to 100,000 are encrypted with all-zero key. Note that the data is encrypted using ECB mode and padding is discarded. The resulting sequence is used for 128-bit testing. Moreover, additional sets of 128, 256 and 512-bit sequences are generated using the iteration $S_i = H(S_{i-1})$ where $S_0 = H(\overline{0})$ and H is the hash function MD5, SHA-2 256 and SHA-2 512 respectively. In this case, the length of $\overline{0}$ is the message block size of the hash function H. Then, the binary representations of the decimal parts of π and $\sqrt{2}$ are tested. For each number, we take as many bits so that 100,000 1024-bit and 4096-bit sequences are generated respectively.

The above mentioned strings measure the behavior of the test on random data. We also generate a *1% weight biased* sequence in order to see if the tests can detect non-random data. For this purpose, using the random number generator of Microsoft .Net Framework, we generate 100,000 128 bit sequences where each bit is 1 with probability 50,5% and 0 with probability 49,5%. The results are given in Table 1. According to the results the first five generators pass all the tests (since all the p values are greater than 0.01) while the biased sequence fails all the tests.

Table 1: Application of randomness tests to different pseudorandom number generators and biased sequences for $N = 100,000$

a	$\sqrt{2}$(4096-Bit)	π(1024-Bit)	SHA-2(256-Bit)	MD5(128-Bit)	AES(128-Bit)	1% Biased(128-Bit)
$y = 0$	0.651536	0.765225	0.723788	0.595462	0.794321	0.000121
$y = 1$	0.261310	0.257546	0.111009	0.785019	0.627966	1.01E-92
$y = -1$	0.862806	0.795376	0.014390	0.495422	0.664016	1.56E-37
$y = 2$	0.176344	0.452462	0.936082	0.728649	0.947433	7.30E-262
$y = -2$	0.631532	0.062998	0.717032	0.864984	0.545166	5.62E-181
$y = 3$	0.708952	0.226330	0.277121	0.788715	0.570326	0
$y = -3$	0.431498	0.127274	0.027632	0.670893	0.519191	4.25E-253
$y = 4$	0.581227	0.780811	0.116235	0.317354	0.452234	0
$y = -4$	0.689942	0.020685	0.097331	0.720714	0.654505	0

8 Conclusion

In this work, we define a family of randomness tests based on random walk statistics. We give recursive formulas that are feasible to compute to obtain the exact probabilities for the number of excursions in a string, namely, the number of strings which intersect the line $y = t$ exactly k times. Moreover, using the exact distributions for all random walk statistics obtained, we introduce a new statistical randomness test suite, RW-9, consisting of 9 tests. Afterwards, we apply the family of these randomness tests to various collections of strings, consisting of accepted as random looking ones and biased ones. The results suggest that the tests defined are all sensitive to both random and non-random data. The sequences generated by $\sqrt{2}$, π, SHA-2 512, SHA-2 256, MD-5 and AES-128 produced p-values greater than 0.01 for all tests, while, biased sequence failed in all 9 tests. As a future work, the correlations and the dependencies of the defined randomness tests will be studied.

References

[1] Maurer U. A universal statistical test for random bit generators. J Cryptol 1992;5:89-105.

[2] L'Ecuyer P, Simard R. Testu01: A c library for empirical testing of random number generators. ACM T Math Software 2007;33(4):22.

[3] Doğanaksoy A, Sulak F, Uğuz M, Şeker O, Akcengiz Z. New Statistical Randomness Tests Based on Length of Runs. Math Probl Eng 2015.

[4] M. Sönmez Turan, A. Doğanaksoy, and S. Boztaş. On independence and sensitivity of statistical randomness tests. In International Conference on Sequences and Their Applications (SETA), 2008.

[5] Sulak F, Uğuz M, Koçak O, and Doğanaksoy A (2017) "On the independence of statistical randomness tests included in the NIST test suite," Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 25: No. 5, Article 15.

[6] Rukhin AL, Soto J, Nechvatal J, Smid M, Barker E, Leigh S, Levenson M, Vangel M, Banks D, Heckert A et. al. A statistical test suite for random and pseudorandom number generators for cryptographic applications Sp 800-22 rev. 1a. Gaithersburg, MD, USA: Booz-Allen and Hamilton Inc Mclean Va, 2010.

[7] Doğanaksoy A, Çalık Ç, Sulak F, Turan MS. New Randomness Tests Using Random Walk. In 2nd National Conference Proceedings; 15-17 December 2006; Ankara, Turkey.

[8] Sulak F, Doğanaksoy A, Ege B, Koçak O. Evaluation of randomness test results for short sequences. In: Carlet C, Pott A editors. Sequences and Their Applications - SETA10 6th International Conference Proceedings; 13-17 September 2010; Paris, France. Berlin:Springer-Verlag, 2010, pp.309-319.

[9] Daeman J, Rijmen V. The Design of Rijndael: AES - The Advanced Encryption Standard. Berlin, Germany:Springer-Verlag Berlin Heidelberg, 2002.

Appendix

Table 2: Bin values and expected probabilities for $n = 128$.

	Bin-1	Bin-2	Bin-3	Bin-4	Bin-5	Bin-6	Bin-7	Bin-8
y=0,1,-1	0-1	2-3	4-5	6-7	8-9	10-12	13-16	17-128
	0.140772	0.138555	0.131984	0.121481	0.107849	0.132083	0.119493	0.107782
y=2,-2	0	1-2	3-4	5-6	7-8	9-11	12-15	16-128
	0.139689	0.137524	0.131103	0.120833	0.107487	0.132098	0.120326	0.110939
y=3,-3	0	1-2	3-4	5-6	7-8	9-11	12-15	16-128
	0.208993	0.134829	0.126409	0.114484	0.099982	0.119954	0.105395	0.089954
y=4,-4	0	1-2	3-4	5-6	7-9	10-13	14-128	-
	0.275146	0.130239	0.120194	0.107125	0.132093	0.12112	0.114083	-
y=5,-5	0	1-3	4-6	7-10	11-128	-	-	-
	0.341299	0.18428	0.155113	0.152257	0.167051	-	-	-

Table 3: Bin values and expected probabilities for $n = 256$.

	Bin-1	Bin-2	Bin-3	Bin-4	Bin-5	Bin-6	Bin-7	Bin-8
y=0,1,-1	0-2	3-4	5-7	8-10	11-13	14-17	18-23	24-256
	0.149262	0.097882	0.140597	0.129088	0.114006	0.124929	0.128544	0.115691
y=2,-2	0-1	2-4	5-7	8-10	11-13	14-17	18-22	23-256
	0.148685	0.145223	0.136791	0.124096	0.108276	0.116979	0.102695	0.117257
y=3,-3	0	1-3	4-6	7-9	10-12	13-16	17-21	22-256
	0.148685	0.145223	0.136791	0.124096	0.108276	0.116979	0.102695	0.117257
y=4,-4	0	1-2	3-5	7-8	9-11	12-15	16-21	22-256
	0.196977	0.09583	0.136356	0.123798	0.108137	0.117033	0.118411	0.103459
y=5,-5	0	1-2	3-4	5-7	8-10	11-14	15-20	21-256
	0.245269	0.094143	0.08975	0.123798	0.108137	0.117033	0.118411	0.103459

Table 4: Bin values and expected probabilities for $n = 1024$.

	Bin-1	Bin-2	Bin-3	Bin-4	Bin-5	Bin-6	Bin-7	Bin-8
y=0,1,-1	0-4	5-9	10-14	15-20	21-27	28-35	36-47	48-1024
	0.124395	0.121977	0.116671	0.129449	0.132296	0.122905	0.127728	0.124579
y=2,-2	0-3	4-8	9-13	14-19	20-26	27-34	35-46	47-1024
	0.124275	0.121863	0.116572	0.12936	0.13224	0.122904	0.127822	0.124965
y=3,-3	0-2	3-7	8-12	13-18	19-25	16-33	34-45	46-1024
	0.124275	0.121863	0.116572	0.12936	0.13224	0.122904	0.127822	0.124965
y=4,-4	0-1	2-6	7-11	12-17	18-24	25-32	33-44	45-1024
	0.124154	0.121749	0.116474	0.129271	0.132184	0.122903	0.127915	0.12535
y=5,-5	0	1-5	6-10	11-16	17-23	24-31	32-43	44-1024
	0.124154	0.121749	0.116474	0.129271	0.132184	0.122903	0.127915	0.12535

Table 5: Bin values and expected probabilities for $n = 4096$.

	Bin-1	Bin-2	Bin-3	Bin-4	Bin-5	Bin-6	Bin-7	Bin-8
y=0,1,-1	0-9	10-19	20-30	31-42	43-55	56-72	73-96	97-4096
	0.124297	0.121591	0.127252	0.1274	0.121105	0.128538	0.124726	0.125092
y=2,-2	0-8	9-18	19-29	30-41	42-54	55-71	72-95	96-4096
	0.124267	0.121562	0.127226	0.127379	0.121093	0.128538	0.124749	0.125186
y=3,-3	0-7	8-17	18-28	29-40	41-53	54-70	71-94	95-4096
	0.124267	0.121562	0.127226	0.127379	0.121093	0.128538	0.124749	0.125186
y=4,-4	0-6	7-16	17-27	28-39	40-52	53-69	70-93	94-4096
	0.124237	0.121534	0.127199	0.127357	0.121081	0.128538	0.124773	0.12528
y=5,-5	0-5	6-15	16-26	27-38	39-51	52-68	69-92	93-4096
	0.124237	0.121534	0.127199	0.127357	0.121081	0.128538	0.124773	0.12528