Alternative Approach to Maurer's Universal Statistical Test

Ali DOĞANAKSOY, Cihangir TEZCAN

***Abstract*—Statistical tests for randomness play an important role in cryptography since many cryptographic applications require random or pseudorandom numbers. In this study, we introduce an alternative approach to Maurer's Universal Test. This approach allows us to test short binary sequences as small as 66 bits and to choose slightly larger block sizes. Moreover, it does not have an initialization part and requires less time to test a binary sequence.**

***Keywords* —Randomness, Statistical Test, Maurer's Universal Test**

I. Introduction

RANDOM numbers have many uses in cryptography such as keystreams of one-time pads, secret keys of symmetric cipher systems, public key parameters, session keys, nonces, initialization vectors and salts. Hence statistical randomness tests are of great importance for cryptographers for testing the security of cryptographic applications.

There are various statistical test suites in the literature such as NIST [2] which is used as an evaluation tool in the AES selection process, Knuth [8], DIEHARD [5], TestU01 [7], and Crypt-X [6], all of which consisting of a number of statistical tests. In this study, we introduce a new approach to Maurer's Universal Test [1] which is one of the 16 tests used in NIST statistical test suite.

Maurer's Universal Test has an initialization part and most of the blocks in this segment has no effect in the test statistic. Moreover, in NIST statistical test suite, block sizes between 6 and 16 are recommended, where the lower bound makes the test inapplicable for binary sequences shorter than $387,840$ bits. However, our approach does not contain an initialization part, it can be used to test short sequences as small as 66 bits and block size can be chosen larger for long sequences. Moreover the new approach requires slightly less time to test a binary sequence.

The outline of the paper is as follows. In section II, we briefly summarize Maurer's Universal Test and introduce our approach in section III. In section IV, we present the advantages of this approach and in the last section, we conclude our paper.

II. Maurer's Universal Test

Maurer's Universal Statistical Test which is introduced by Ueli M. Maurer in 1992 in [1] measures per-bit entropy of a stream which is considered as the correct quality measure for a secret key source in a cryptographic application. The test is designed to detect any one of the very general class of statistical defects that can be modeled by an ergodic source with finite memory M where M is less than or equivalent to the block size L.

Let $\{a_n\} = a_1, a_2, \ldots, a_N$ be a binary sequence of length N. To apply the test, we partition this sequence into adjacent non-overlapping blocks of length L and compute the integer values of these blocks and obtain a new sequence $\{t_n\} = t_1, t_2, \ldots, t_k$ where $k = \lfloor \frac{N}{L} \rfloor$ and $t_i \in \{0, 1, 2, \ldots, 2^L - 1\}$. The remaining bits at the end of the sequence are discarded.

The first Q blocks of $\{t_n\}$ is called the initialization part and the remaining K blocks are called the test part where $K+Q = \lfloor n/L \rfloor$. For every block in the test part, we calculate the distance of that block to its previous occurrence. That is, the distance r for the block t_i means t_i and t_{i-r} are the same and the integers between them are different than t_i. These calculations can be done efficiently by scanning the sequence $\{t_n\}$ once and storing the places of the last occurrences of blocks in an array of size 2^L. If we denote these distances by c_i, the test statistic f_n is

$$f_n = \frac{1}{K} \sum_{1}^{K} \log_2 c_i. \tag{1}$$

The reference distribution for the test statistic is the half-normal distribution. The p-value is obtained as follows:

$$c = 0.7 - \frac{0.8}{L} + \left(4 + \frac{32}{L}\right) \frac{K^{-3/L}}{15} \tag{2}$$

$$\sigma = c\sqrt{\frac{variance(L)}{K}} \tag{3}$$

$$p - value = erfc\left(\left|\frac{f_n - expectedvalue(L)}{\sqrt{2}\sigma}\right|\right). \tag{4}$$

If the obtained p-value is less than the probability of type I error, which is a small value between 0.01 and 0.001, we assume that the sequence is obtained from a non-random resource.

An efficient and clear implementation of the complementary error function $erfc$ can be found in [3]. The values of variances and expected values are given in Table IV at Appendix A.

Ali Doğanaksoy is with the Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University, Ankara, 06531 TURKEY e-mail: (aldoks@metu.edu.tr).

Cihangir Tezcan is with the Institute of Applied Mathematics, Middle East Technical University, Ankara, 06531 TURKEY e-mail: (forgottenlance@gmail.com).

III. The New Approach

Let $\{a_n\} = a_1, a_2, \ldots, a_N$ be a binary sequence of length N. By partitioning this sequence into adjacent non-overlapping blocks of length L and computing the integer values of these blocks, we obtain a new sequence $\{t_n\} = t_1, t_2, \ldots, t_k$ where $k = \lfloor \frac{N}{L} \rfloor$ and $t_i \in \{0, 1, 2, \ldots, 2^L - 1\}$. The remaining bits at the end of the sequence are discarded. Moreover, from $\{t_n\}$ we obtain yet another sequence $\{c_n\}$ where c_i is the distance between the integer t_i and its next occurrence. If the integer t_i is its final occurrence in the sequence $\{t_n\}$, we assign the value 0 to c_i. Thus the sequence $\{c_n\}$ also has length k. Note that in this approach we do not use an initialization part. Hence when calculating the distances, considering the previous occurrence of a block, instead of its next occurrence does not change the test statistic.

If c_i is r for some i, this means that t_i and t_{i+r} are the same integers and the integers between them are different than t_i. The probability of such a situation is

$$prob(c_i = r) = \frac{(2^L - 1)^{r-1}}{(2^L)^r}. \quad (5)$$

Thus the probability for $c_i = 0$ is obtained in the following way:

$$prob(c_i = 0) = 1 - \sum_{i=1}^{k} \frac{(2^L - 1)^{i-1}}{(2^L)^i}. \quad (6)$$

Note that since t_k is the last element of the sequence $\{t_n\}$, c_k must be 0. Similarly c_{k-1} is either 0 or 1. Thus a distance r in the sequence c_i can be observed only in $k - r$ different places. Hence the expected number of appearance of the value r in the sequence $\{c_n\}$ is

$$E(r) = (k - r)\frac{(2^L - 1)^{r-1}}{(2^L)^r}. \quad (7)$$

Let d_i denote the number of appearance of the value i in the sequence $\{c_n\}$. We apply the test by calculating d_i's and performing χ^2 of goodness of fit test to d_i and $E(i)$ values.

The number of appearance of 0 in the $\{c_n\}$ sequence is equivalent to the number of distinct integers in the $\{t_n\}$ sequence. To apply the test, we require that every integer in the set $\{0, 1, \ldots, 2^L - 1\}$ should appear in $\{t_n\}$ at least once. Thus we will consider the cases when the probability of every possible integer values not appearing in $\{t_n\}$ is less than 10^{-4}. Hence to perform the test, the largest block size L that can be chosen for a $\{t_n\}$ sequence with length k is the largest L value satisfying the following equation:

$$1 - \left(\frac{2^L - 1}{2^L}\right)^k - \left(\frac{2^L - 2}{2^L}\right)^k - \ldots - \left(\frac{1}{2^L}\right)^k > 0.9999 \quad (8)$$

The largest L values satisfying the above inequality are the suggested block sizes and are given in Table I.

We apply χ^2 of goodness of fit test to obtain p-values. The degree of freedom d is $k - 1$ and χ^2 value is

$$\chi^2 = \sum_{i=1}^{k} \frac{(d_i - E(i))^2}{E(i)} \quad (9)$$

TABLE I
Largest Possible Block Sizes

Sequence Length	Block Size
$66 \leq n \leq 206$	2
$207 \leq n \leq 571$	3
$572 \leq n \leq 1,454$	4
$1,455 \leq n \leq 3,509$	5
$3,510 \leq n \leq 8,224$	6
$8,225 \leq n \leq 18,839$	7
$18,832 \leq n \leq 42,407$	8
$42,408 \leq n \leq 94,269$	9
$94,270 \leq n \leq 207,448$	10
$207,449 \leq n \leq 452,663$	11
$452,664 \leq n \leq 980,823$	12
$980,824 \leq n \leq 2,112,599$	13
$2,112,600 \leq n \leq 4,257,059$	14
$4,257,060 \leq n \leq 9,657,775$	15
$9,657,776 \leq n \leq 20,522,858$	16
$20,522,859 \leq n \leq 43,460,243$	17
$43,460,244 \leq n \leq 91,749,460$	18
$91,749,461 \leq n \leq 193,156,859$	19
$193,156,860 \leq n \leq 405,629,468$	20
$405,629,469 \leq n \leq 849,890,425$	21
$849,890,426 \leq n \leq 1,777,043,709$	22

and we use the complement of incomplete gamma function $gammq$ (see [3] for an efficient and clear implementation) to obtain the p-value where

$$p - value = gammq\left(\frac{d}{2}, \frac{\chi^2}{2}\right) \quad (10)$$

If the obtained p-value is less than the probability of type I error, which is a small value between 0.01 and 0.001, we assume that the sequence is obtained from a non-random resource.

Since we assumed that every possible integer values is observed in the sequence $\{t_n\}$, before testing a binary sequence, we check that if the corresponding $\{t_n\}$ sequence is as we desired. If it is not, the test will not be applied to that binary sequence.

IV. Comparison

In Maurer's Universal Test, the initialization segment contains $10 \cdot 2^L$ bits and most of the blocks of this part have no effect in the test result. This is due to the fact that only the place of the last occurrence of an integer in the initialization part is considered in the test segment. However, there is no initialization part in the presented method which allows us to test the whole sequence without wasting any parts of the sequence.

Another important difference between the two methods is the length of the sequences that can be tested. In Maurer's Universal Test, sequences which are shorter than $387,840$ bits cannot be tested. However, our approach can be applied to test sequences as short as 66 bits.

It is important to notice that Maurer's Universal Test requires a long sequence with initialization part of $10 \cdot 2^L$ blocks and test part of $1000 \cdot 2^L$ blocks which makes the test not suitable for block size length larger than 16. In our new method, the block size can be taken larger than 16 for long sequences but notice that the distances are stored in an array of size 2^L in both methods. Hence the test becomes infeasible for large L.

Yet another difference is the speed of the both algorithms. Since there are no logarithms involved in our method, it requires less time to test a binary sequence. We measured the speed of the tests by testing 1000 binary sequences of length 800000 bits on a PC with an Athlon 64 3700+ processor at 2.2 GHz, 3 GB of RAM and running Windows XP. These sequences are taken from [5] and we used the same block size for both of the tests. Results are given in the Table II.

TABLE II
THE TIME COMPARISON OF THE TWO TESTS BASED ON 1,000 RANDOM SEQUENCES OF LENGTH 800,000 BITS

Maurer's Universal Test	102 seconds
New method	87 seconds

The linear relationship between the two tests can be seen from Pearson product-moment correlation coefficient [4], which is a value between -1 and 1. Correlation between the variables gets stronger when the coefficient comes closer to -1 or 1. However, the correlation coefficient 0 does not mean that the variables are uncorrelated since it measures only the linear relationship. By using the p-values we obtained from the speed comparison of the tests, we calculated Pearson's coefficient as 0.3.

V. CONCLUSION

In this study, we proposed an alternative method for applying Maurer's Universal Statistical Test. The comparison of this approach and Maurer's Universal Test is made in Section IV. Although the idea behind the both methods are the same, the differences in the test structure (especially the initialization part of Maurer's Universal Test) result in different p-values for a binary sequence.

Our approach does not use an initialization part and it can be used to test short binary sequences. It allows us to work with larger block sizes and it requires less time to test a binary sequence. The linear relationship between the two tests can be seen from Pearson product-moment correlation coefficient which is calculated in section IV.

APPENDIX A
MAURER'S UNIVERSAL TEST PARAMETERS

The number of blocks used in the initialization part of Maurer's Universal Test and the block sizes are given in Table III. The expected values and variances are given in Table IV.

TABLE III
PARAMETERS IN NIST STATISTICAL TEST SUITE

n	L	$Q = 10 \cdot 2^L$
$\geq 387,840$	6	640
$\geq 904,960$	7	1,280
$\geq 2,068,480$	8	2,560
$\geq 4,654,080$	9	5,120
$\geq 10,342,400$	10	10,240
$\geq 22,753,280$	11	20,480
$\geq 49,643,520$	12	40,960
$\geq 107,560,960$	13	81,920
$\geq 231,669,760$	14	163,840
$\geq 496,435,200$	15	327,680
$\geq 1,059,061,760$	16	655,360

TABLE IV
EXPECTED VALUES AND VARIANCES

L	expectedvalue	variance
6	5.2177052	2.954
7	6.1962507	3.125
8	7.1836656	3.238
9	8.1764248	3.311
10	9.1723243	3.356
11	10.170032	3.384
12	11.168765	3.401
13	12.168070	3.410
14	13.167693	3.416
15	14.167488	3.419
16	15.167379	3.421

ACKNOWLEDGMENT

The authors would like to thank Meltem Sönmez Turan for her helpful comments.

REFERENCES

[1] U. M. Maurer, *A Universal Statistical Test for Random Bit Generators*, Journal of Cryptology. Vol. 5, No. 2, 1992, pp. 89-105.
[2] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, 1992.
[3] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Edition, ISBN 0521431085, Cambridge University Press, 1992.
[4] J. L. Rodgers and W. A. Nicewander, *Thirteen Ways to Look at the Correlation Coefficient*, The American Statistician, Vol. 42, No. 1 (Feb., 1988), pp. 59-66.
[5] G. Marsaglia, *DIEHARD Statistical Tests*, http://stat.fsu.edu/pub/diehard/.
[6] Information Security Institute, *Crypt-X*, 1998, http://www.isi.qut.edu.au/resources/cryptx/.
[7] P. L'Ecuyer and R. Simard, *TestU01: A C Library for Emprical Testing of Random Number Generators*, ACM Transactions on Mathematical Software, 33, 4, Article 22, August 2007.
[8] D.E. Knuth, *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*, Addison-Wesley, 1981.