True Random Number Generation Based on Environmental Noise Measurements for Military Applications

N.G. BARDIS [1,2], A.P. MARKOVSKYI [3],
N. DOUKAS [1,4], N. V. KARADIMAS [1,4]
University of Military Education,
[1]Hellenic Army Academy, [2]Hellenic Naval Academy, [4]Hellenic Air Force Academy
Department of Computer Sciences,
[1]Vari - 16673, [2]Terma Hadjikyriakou Avenue, Piraeus - 18539, [4]Dekelia Air Base, Tatoi, 14451, GREECE
[1,2]bardis@ieee.org, [1,4]nd@aeub.gr, [1,4]nkaradimas@sse.gr
[3]Department of Computer Engineering, National Technical University of Ukraine,
37, Peremohy, pr. Kiev 252056, KPI 2003, UKRAINE
[3] markovskyy@mail.ru

Abstract: - True random number generators can significantly contribute to the development of high security cryptographic schemes, such as those required for use in military applications. This article presents some the results of an innovative method for the generation of truly random number sequences, based on environmental noise measurements. The statistical properties of different noise types have been studied. Based on this study, an efficient random number generator has been developed that uses signals from the built in microphone that is ubiquitous most current personal computers and other personal information processing systems. Statistical measures have been determined that measure the randomness qualities of the output sequence. These measures have been studied for different input noise properties. The merit of the proposed generator in terms of the rate at which random numbers are produced has also been assessed.

Key-Words: - True Random Number Generation, Cryptography, Information Security.

1 Introduction

Random numbers are widely used in information processing and communication systems for a variety of applications. The most common such applications are system modeling and simulation, digital device diagnostic systems, implementation of computational solutions for mathematical problems using the Monte Carlo method, creation of digital sound and video, as well as in information security [1], [2], [3] and military systems [4]. For the majority of these applications pseudo random number sequences (PSRNG) are used instead of random numbers. PSRNG are generated by software that produces sequences that may be assumed to be random. However, in many cases these assumptions are not valid, since the appearance of a number determines the next number that is going to appear in the sequence.

This work was supported by a grant by the Greek Ministry of Development - General Secretariat for Research and Technology and the European Community Social Funds. The main part of this work was carried out when the authors where attending the Operational program "PENED 2003"

For a significant part of the modeling problems, it is highly important that there is no connection between the numbers in a sequence. It is necessary that after the appearance of code A, the next code that appears depends only on the statistics of the underlying distribution and may not deterministically identified. Hence code A may reappear with equal probability as any other code (for the case of the uniform distribution) [5].

Similar problems appear when PSRN sequences are used in diagnostic tests for digital devices. The nature of this sequence has as a consequence the fact that the diagnostic procedure examines only a limited number of track tests and has therefore a limited ability for identifying design of manufacturing faults.

Similar problems exist when using PSRNG for simulation and information security systems. Research into the dynamics of true random number generators (TRNG) for use in information systems is ongoing [1], [5] The reason is that a large number of applications may significantly benefit by the availability of independent random number inputs. Therefore, the development of innovative and efficient TRNGs is an important and urgent prerequisite for current information system

development. In this paper, an innovative approach to TRNG is presented that is based on using standard hardware and limited post processing in order to obtain the random bits. The method uses ambient noise measurements as a source of randomness. The performance of the generator and the quality of the random sequence are examined for different noise types via quantitative observations. Targets for further study on the front of TRNG are identified.

2 Existing TRNG methods

The importance of generating true random number sequences is the driving force for the research into the creation of efficient TRNGs. The concept of efficiency is defined based on the requirements of each implementation and these requirements are used to determine efficiency measures for the quality evaluation of each generator. It is however possible to state some general quality principles that may be used in most practical cases. These principles are outlined below, based on the uniformly distributed random number sequence case. Other distributions may be obtained as a function of the uniform ones [6].

1. Statistical criteria for the quality of random numbers assume equal probabilities for the appearance of any number, independently of the previously observed ones. All samples of the sequence must therefore be Independent, Identically Distributed (IID) random variables [6]. Additionally, all samples must obey the same probability distribution function (uniform, normal etc).
2. The non – predictability principle of the generated number implies that it is not possible to predict their sequence.
3. Productivity criteria; the random number generator must be capable of producing numbers at a rate that is sufficiently high for the requirements of the application for which it is intended.
4. Costs of any additional hardware and software that will be used for the generation of the random number sequence. This cost will depend on the computational platform used in order to do the generation. The platform will often be a personal computer. However the current trend for information system development is the increasing use of mobile information processing equipment, such as mobile phones with processing capabilities

3 Improvement on existing methods

For a large number of applications, especially for those intended for use in for mobile devices, the use of dedicated, specialised hardware for RNG is not feasible, due to cost, volume and power consumption limitations. In such cases, the ability to generate truly random numbers with the readily available hardware acquires an additional significance. At the same time, it is important to note that this significance increases with the increased use of mobile platforms as end user, terminal computational devices in computer networks [7]. The connection of such devices to networks requires the implementation of the necessary protocols and security algorithms, which in turns requires the generation of truly random numbers. Based on their use of additional hardware, existing approaches for TRNG may be classified into three groups:

1. Acquisition of the random numbers by software that uses non random parameters of the computing system and the control and the reconstruction of the sequence is restricted by technological difficulties. This group of TRNG includes algorithms that are based on measuring time intervals, the calculation of the signatures of the digital data that exist in memory, via the use of the processor parameters [1]. The main drawback of this approach is the possibility that the sequence produced using such method may be predicted and reproduced. The use of such generators for is hence limited for information security applications. A characteristic example of such an algorithm may be found in [8]. The method proposed uses the signature of the data in RAM, in order to acquire the randomness necessary.
2. Acquisition of the random numbers by measurement of parameters of random natural processes that are taking place in specialized hardware connected to the computing system. Such processes include natural processes that appear regularly but are random. It is often the case that for this purpose, the function of unstable electronic circuits is used in practice [1]. A less common approach is the use of measurements of parameters of the radioactive decomposition and of quantum processes [1]. The main drawback of these approaches in the need for additional hardware as well as the complexity added for interfacing these external devices and the main system. Other examples

of this class of algorithms may be found in [2], [4], [9], [10].

3. The acquisition of random numbers by measurement of parameters produced by the operation of hardware already incorporated in the computational platform. Hardware that is suitable for this purpose includes:
   - Mouse or joystick
   - Keyboard
   - Hard disk
   - Microphone
   - Webcamera

The first two devices can only produce random numbers at low rates and are hence limited in their applicability. A very effective method for forming truly random numbers uses the measurement of the speed of rotation of the hard disks, which depends on the naturally random process associated with turbulence phenomena in the air surrounding the rotating part of the hard disk drive. A practical implementation of this idea may be easily obtained using software and measuring some characteristic properties of the disks. This method however, cannot be used for TRNG in multiprogramming computer systems or computer systems with limited resources, such as PDAs.

For the majority of applications, the most appropriate source of randomness and hence the recommended method for TRNG, that is used by the majority of applications, is ambient noise measurement via the built in microphone. This device is a standard part of modern personal computers, laptop computers, PDAs and of course mobile phones. Ambient noise that is the sum of the noises originating from a variety of sources, that are truly random and cannot be reproduced. For the practical use of the ambient noise algorithm, it is necessary to study the statistical properties of the microphone signals and the noise, the process of transferring data from the microphone and the rate at which random numbers may be obtained. Example algorithms for true random number generation using built in computer hardware can be found in [11], [12] and [13]

4 Structure of an ambient noise measurement based TRNG.

The acquisition of a random numbers using the built in microphone, demand the use of a series of transformations and results in a sequence of uniformly distributed random numbers. These random numbers are readily useable or may be transformed using one of the standard transformations [6] to obey any necessary probability distribution function. The structure of a TRNG that implements the above methodology is shown in Figure 1.

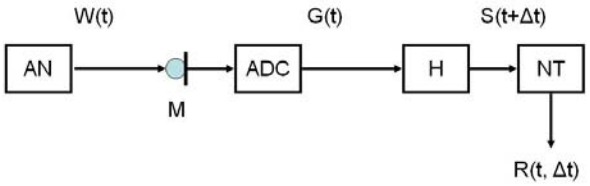

Figure 1: TRNG based on ambient noise measurements.

The ambient noise (AN), represented as the analogue signal W(t), is captured by the microphone M. The noise signal is converted to the discrete domain via the analogue to digital converter (ADC), resulting in the signal G(t). If the transformation performed at the microphone is defined to be M(t), the signal G can be expressed as:

$$G(t) = M\left(W(t)\right) \tag{1}$$

The discrete signals are buffered in the memory H in digital format, creating packets. The length N_S of the buffer covers a period Δt such that $\Delta t = N_S \cdot \tau$, where τ is the sampling period. Hence the packet formed contains:

$$\Delta t : S(t+\Delta t) = \left\{M(W(t)), M(W(t+\tau), M(W(t+2\tau)..., M(W(t+\Delta\tau))\right\} \tag{2}$$

The packet S(t, Δt) is subsequently subject to a normalization transformation (NT) that creates the packet of the uniformly distributed random variables R(t,Δt). If the normalization function is denoted by F(X), then the resulting sequence R(t,Δt) may be expressed as:

$$R(t,\Delta t) = F(S(t,\Delta t) = F(M(W(t)), M(W(t+\delta)),...M(W(t+\Delta t))) \tag{3}$$

If the size of the resulting packet R(t,Δt) of the acquired numbers is V_R bits, for these numbers to be truly random the overall entropy H_R of the packet must also be V_R (or tend to V_R). The proof of this statement will be presented in a future publication. A measure of randomness can hence be calculated as the ratio:

$$D_R = \frac{H_R}{V_R} \tag{4}$$

This measure characterizes the randomness of the packet; the closer to 1 D_R becomes, the better the sequence R(t,Δt) approximates the randomness properties required. A complete theoretical analysis

and proof for this statement will again be presented in a future publication.

It is apparent that the value of the entropy H_R depends on the value of the input signal entropy H_S of the signal S(t, Δt) and on the form of the normalization function F(X). This observation imposes some restrictions on the transformation function F(X). Firstly, the input entropy H_S or the input signal S(t, Δt) must be maintained. Additionally, it should be noted that the entropy H_S will inevitably be larger than the entropy H_R of the resulting sequence, or $H_S \geq H_R$.

The entropy H_S of the packet is obtained from the calculation of S(t, Δt). The size V_S of this packet is calculated in bytes. The entropy of one byte of the discretized noise is h_S. If the size of the sample that is obtained from the ADC is k, then the following relationship is valid:

$$H_S = V_S \cdot h_S = N_S \cdot k \cdot h_S = \frac{\Delta t \cdot k \cdot h_S}{\tau} \quad (5)$$

The above expression is valid only for a certain constant value of τ, since for low values of τ, the average value of the entropy of the noise byte entropy depends on τ; the smaller τ is, the smaller h_s becomes. With the constant value of the discreteness period τ, the average value of the entropy h_S is only determined by the type of the sampled noise (e.g. parasitic environment noise, human speech, music). The proof of this statement will be given in a future publication.

The equation given above implies that with constant values for the size of the resulting packet R(t,Δt) of random numbers, for the discreteness period τ of the packet S(t, Δt), for the buffering period Δt and consequently for the rate of generation of random numbers, the entropy of the output will depend only on the type of ambient noise present. The aim of the research is therefore to determine the values of the mean h_S for different types of ambient noise. Another aim of the research is to determine the nature of distribution the random event so that the buffering period for S(t, Δt) can be sensibly selected.

5 Experimental Results

This section presents the results of initial studies on the characteristics of the generated random signals for different ambient noise types.

For the analysis, the ADC built in to the computer was used. The environment sounds captured included office silence (only noise from computers, from air conditioning and from outside the office being heard), single person speaking and the cocktail party situation (many people speaking at the same time combined with multimedia noise). The sounds were captured at 22050 Hz and 16 bits. Random bit sequences of 20000 random bits each were generated and their statistical properties were examined.

The tests performed on each sequence were six, namely:

1. Entropy measurement
2. Autocorrelation estimate
3. Monobit test
4. Poker test
5. Runs test
6. Long runs test

The last four tests are tests prescribed by the FIPS 800-22 special publication [7]. The first test checks the entropy of the produced sequence, conformant to the analysis given in a previous section. The run is considered to have passed the test, if the ratio D_R defined above is greater than 0.995. The second test produces both a measurable and a visual confirmation that the data does not suffer from any periodic components. The third test checks the observed probabilities for the bit sequence produced and decides if it is reasonably close to the theoretically expected values. The fourth test assembles the bits produced into multi-bit words and checks that the values of these words exhibit the randomness necessary. The Runs test conforms that the repeated appearances of the same bit in the data does not exceed what is theoretically expected while the Long runs test examines if very long runs appear more often than expected. It should be noted that failure of one sequence in one or more tests does not imply that the data is not random. On the contrary, depending on certain conditions under investigation, this observation may confirm the randomness of the process, since it is observed that low probability events appear with the frequency that is theoretically expected.

The success rates observed for the numerical tests (1, 3, 4, 5, 6) are tabulated below for four cases of noise conditions:

1. Single person speaking continuously in an office environment
2. Office noise consisting of air-conditioning fan and operational laptop computer
3. Cocktail party noise with multiple conversations taking place and background multimedia
4. Mixed noise (office noise, multimedia sounds and human conversations)

Table 1: Test results for single person speech

No	Test	Success rate (%)
1	Entropy	100

3	Monobit	99
4	Poker	99
5	Runs	100
6	Long runs	100

Table 2: Test results for office noise

No	Test	Success rate (%)
1	Entropy	100
3	Monobit	100
4	Poker	100
5	Runs	99
6	Long runs	100

Table 3: Test results for cocktail party noise

No	Test	Success rate (%)
1	Entropy	100
3	Monobit	100
4	Poker	100
5	Runs	100
6	Long runs	100

Table 4: Test results for mixed noise

No	Test	Success rate (%)
1	Entropy	100
3	Monobit	100
4	Poker	100
5	Runs	100
6	Long runs	100

Sample observed autocorrelation functions for the produced bit sequences are shown in the figures below. For calculating these autocorrelation functions, Matlab built in, biased estimator was used. The values were normalized and the graphs shown below correspond to non-contiguous sound periods.

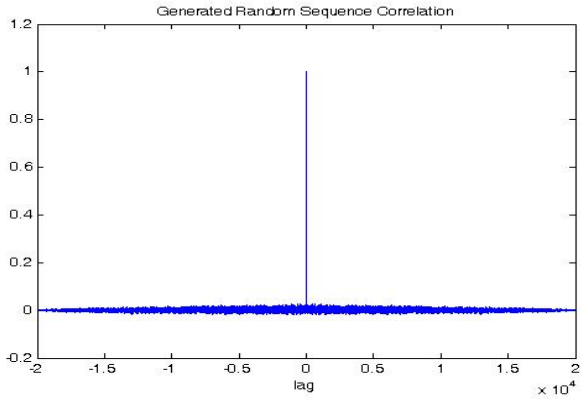

Figure 2: Typical sequence autocorrelation for single person speech

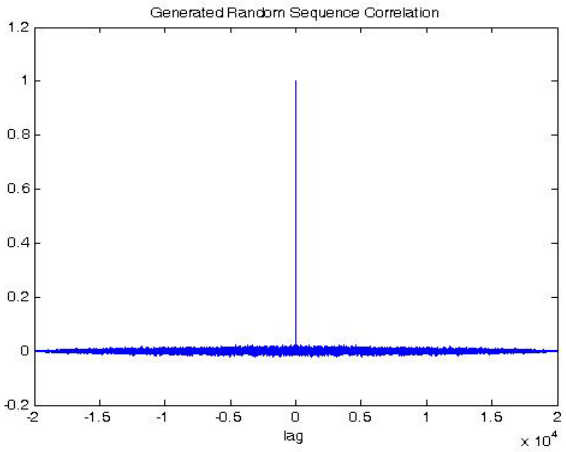

Figure 3: Typical sequence autocorrelation for office noise

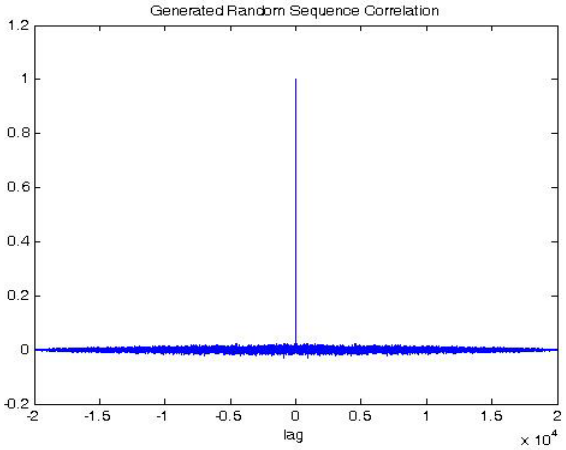

Figure 4: Typical sequence autocorrelation for cocktail party noise

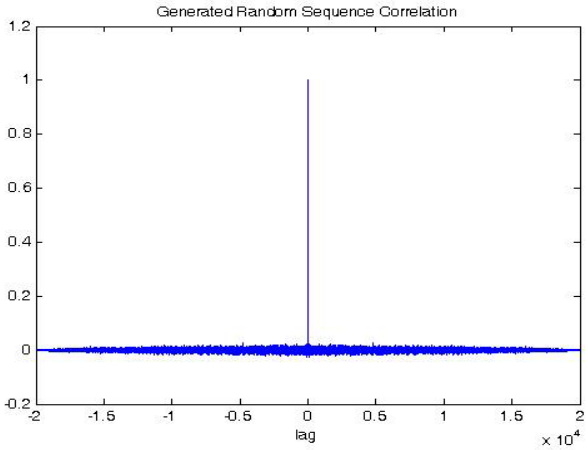

Figure 5: Typical sequence autocorrelation for mixed noise

The above results shown indicate that measurable tests, as adopted by standards bodies [7] indicate that the generated sequences can be considered as truly random bits. These results are strong indication of the quality of the generated sequence.

It has already been pointed out that another significant factor that determines the value of a TRNG is the quantity or the rate at which random numbers are generated. With the 22050 Hz sampling rate, the rate of random number generation was approximately 2.7 KB/sec.

5 Ongoing Work

Research on the analysis of the proposed TRNG is still ongoing, both in the theoretical and the measurement fronts.

From the theoretical analysis presented, future publications are going to present analytical proofs for the entropy requirements for the random sequences produced and the consequent entropy test that was proposed and used earlier on. Additionally, the choice of the optimal buffering period τ will be analysed and theoretically proven.

On the practical application of the idea, the correlation between test results and noise types is still being studied and results on more noise types will be presented. The quality of the produced sequences will need to be confirmed by further tests, both from the FIPS 800-22 suite [7] and other sources [14]. Finally, experiments are ongoing to determine the maximum rate at which the proposed algorithm may produce random numbers. This maximum will also be investigated in terms of theoretical considerations regarding the type of the ambient noise present in each case.

6 Conclusion

The problem of true random number generation is examined. A technique is proposed that does not require the use of specialized hardware. Initial experiments have been conducted and an initial analysis of both theory and results has been carried out. The studies have shown the results to be promising and the aspects of the overall proposal that need further studying have been identified.

References:

[1] Menezer A.J., Van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptograp¬hy. CRC-Press, 1997.- 780 c.

[2] P. Stavroulakis, Chaos Applications in Telecommunications. CRC Press, 2006.

[3] P. Stavroulakis. *TErrestrial Trunked RAdio – TETRA: A Global Security Tool*. Springer Verlag 2007.

[4] Blaszczyk, Marta; Guinee, Richard A.; A novel modelled true random binary number generator for key stream generation in cryptographic applications. Military Communications Conference, 2008. MILCOM 2008. IEEE 16-19 Nov. 2008 Page(s):1 – 7

[5] Morgan D.R. Analysis of Digital Random Number Generated from Serial Samples of Correlated Gaussian Noise. // IEEE Transaction on Information Theory, Vol.27, № 2, 1981 – pp. 235-239.

[6] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* Fourth Edition, McGraw Hill 2002

[7] A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. *National Institute for Standards and Technology Special Publication 800-22.* Revision 1, August 2008

[8] Daniel E. Holcomb, Wayne P. Burleson, and Kevin Fu. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. IEEE Transactions On Computers, Vol. 57, No. 11, November 2008

[9] A Robust Random Number Generator Based on a Differential Current-Mode Chaos Katz, O.; Ramon, D.A.; Wagner, I.A.; Very Large Scale Integration (VLSI) Systems, IEEE Transactions on Volume 16, Issue 12, Dec. 2008 Page(s):1677 – 1686

[10] S. Joshi. Addressing the physical security of encryption keys. *Defense Electronics,* December 2007

[11] Sergio Callegari, Riccardo Rovatti, and Gianluca Setti. Embeddable ADC-Based True Random Number Generator for Cryptographic Applications Exploiting Nonlinear Signal Processing and Chaos. IEEE Transactions On Signal Processing, Vol. 53, No. 2, February 2005.

[12] Jovan Dj. Golic. New Methods for Digital Generation and Postprocessing of Random Data. IEEE Transactions On Computers, Vol. 55, No. 10, October 2006

[13] Nejati, H.; Beirami, A.; Massoud, Y., A realizable modified tent map for true random number generation Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on 10-13 Aug. 2008 Page(s):621 – 624.

[14] Karlheinz Fleischer, Two tests of pseudo random number generators for independence and uniform distribution Lehrstuhl für Statistik und Ökonometrie, Wirtschafts und Sozialwissenschaftliche Fakultät, Universität Erlangen-Nürnberg, Nürnberg, Germany July 1995.