computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

SOURCES OF RANDOMNESS FOR USE IN RANDOM NUMBER GENERATION

A. G. Fragopoulos[1)] and D. N. Serpanos[2)]

[1), 2)] Dept. of Electrical and Computer Engineering
University of Patras
GR – 26504 Patras, GREECE
{afragop, serpanos}@ee.upatras.gr

Abstract: *Efficient generation of random numbers plays significant role in cryptographic applications. Such a generator has to produce unpredictable and un-correlated random bits. Random number generators are classified as pseudo-random number generators (PRNGs) and true random number generators (TRNGs). The first ones have the disadvantage that they can be proven predictable, while the latter ones can produce true random bits but it is not easy to re-produce specific sequences or implement them in constrained environments and there may exist correlations and biases of produced sequences. A third class of random number generators has been introduced, called hybrid-random number generators (h-RNGs), where there is a combination of a cryptographically strong PRNGs or TRNGs which are seeded, and possibly re-seeded, through a source of randomness with high entropy. In this paper, we present an overview of various sources of randomness that can be used either as direct random number generators or as seed sources in h-RNGs, for application in embedded systems.*

Keywords: – *Random Number Generators, randomness, embedded systems, RNGs, true RNGs, hybrid RNGs, cryptographic RNGs.*

1. INTRODUCTION

Nowadays, more and more applications, such as e-commerce, e-banking, military services and, in general, applications that involve handling of sensitive data that might be compromised by an adversary, use unsafe media, like the Internet, to interchange these data. Protection of data can be achieved using various cryptographic methods and protocols. Use of cryptography can achieve basic security requirements, such as privacy, confidentiality and integrity of user data. Efficient generation of random numbers plays significant role in cryptography. Various cryptographic primitives require generation of cryptographically strong random numbers, i.e. digital signature schemes, public-key algorithms like RSA and ECC, and symmetric encryption and decryption algorithms. The one-time pad is considered the perfect cryptosystem, under the assumption that it is provided with a truly random bit stream. Furthermore, various cryptographic protocols like SSL are based on generation of cryptographically strong random numbers; Netscape's SSL implementation was hacked due to failure to generate strongly random bits [33]. In general, random number generators (RNGs) are classified as *pseudo-random number generators* (P-RNGs) and as *true random number generators* (T-RNGs). In the first class, the sequence is produced by a computer using a deterministic algorithm which takes an initial value –seed— and produces random numbers. The entropy of the produced sequence is always smaller or equal to the entropy of the seed, so it is necessary to use as seed, data with high entropy. In the second class, the sequence is produced by a physical source of randomness, like radioactive decay, noise, etc. There is also a third class of random number generators, called *hybrid-RNGs*, which use a T-RNG as seed generator and expand it. Clearly, the requirements of the different generators are varying; thus, it is necessary to identify sources of randomness that can be used for deployment of random number generators in various application areas. Our interest is to identify appropriate RNGs for adoption in embedded, constrained environments, such as mobile appliances, PDAs, etc.

In this paper we present an overview of various sources of randomness which can be used in direct random number generators or as seed in hybrid random number generators. The paper is organized as follows. Section 2 presents a definition of randomness. Section 3 briefly presents the properties

of cryptographic random number generators and also describes various statistical tests and methods for evaluation of randomness. Section 4 describes briefly various methods (de-skewing techniques) that have been proposed in order to eliminate biases and correlations that may appear in raw data obtained from a source. Section 5 describes the sources of randomness, indicating the application areas (embedded or general-purpose), where each source is suitable.

2. DEFINITION OF RANDOMNESS

Randomness constitutes a fundamental, but at the same time a not well defined, notion in mathematics and physics. In 1919, von Mises [1] defined a random sequence as an infinite sequence of binary digits, in which, asymptotically, the number of 0s is a finite fraction of the total bits and when the same fraction is found for every infinite subsequence that can be chosen independently. That claim proved to be inadequate and to have various problems [2]. Wolfram [3] defined randomness through the notion of redundancy, i.e. we can consider a sequence of numbers random, if we cannot identify the existence of patterns, at least with conventional technology and methods of recognition. The key for defining randomness is the notion of non-compressibility of an object, which in turn means that we cannot provide a more compact representation of the object. In our effort to generate random numbers for practical purposes, we use computers which are deterministic finite state machines with quite predicted output for given input. Chaitin [4, 5] gave an algorithmic definition of randomness, based on the capabilities and constraints of computing systems. A sequence of bits can be considered random, if the minimal algorithm that can produce the sequence has the same bits of information with the sequence itself. We may define the complexity $A_C(s)$ of a sequence s of binary digits, as the length (in bits) of the minimal program p that will output the sequence s when it is executed on computer C, i.e.

$$A_C(s) = \min_{C(p)=s} \log_2(p) \quad (1)$$

A binary sequence s can be considered random, when its complexity is almost equal with its length (in bits), i.e.

$$A(s) \approx \log_2(s) \quad (2)$$

Claude Shannon, a pioneer in information theory, defined randomness of a binary sequence through the amount of information that is contained in the sequence [6, 7]. Considering a random variable X that takes values on a finite set $x_1, x_2, ..., x_n$, we define the entropy H(X) of X, as

$$H(X) = -\sum_{i=1}^{n} \Pr[X = x_i] \cdot \log_2(\Pr[X = x_i]) \quad (3)$$

The entropy quantifies the amount of information that is contained in an observation of the random variable X. Consider, for example, a source that generates binary sequences of length s, i.e. $s \in \{0,1\}^n$. If the source is random, then each sequence should be generated with the same probability $p_i = 1/2^n$, where $p_i = \Pr[S = s_i]$. The entropy of random variable X can be calculated by (3), giving

$$\begin{aligned} H(X) &= -\sum_{i=1}^{N} p_i \log_2 p_i = \\ &= -(p_1 \log_2 p_1 + ... + p_N \log_2 p_N) = \\ &= -p \log_2(p^N) = \\ &= -\log_2(p^{pN}) = \\ &= -\log_2\left(p^{\left(\frac{1}{2^n} \cdot 2^n\right)}\right) = -\log_2\left(\frac{1}{2^n}\right) = \\ &= n \end{aligned}$$

Thus, the entropy of a truly random binary sequence equals with its size in bits.

3. EVALUATING RANDOMNESS

In our work, we want to identify some sources of randomness that can be used to produce cryptographically strong random bit sequences. The properties that should be fulfilled by a random number generator depend on their use, e.g. random numbers for stochastic processes like Monte Carlo simulation and cryptographic random numbers have different objectives and need to have different properties. The main property that should characterize a cryptographic random number generator is the ***computational unpredictability*** of successive bits of the sequence. This means that, for such a generator if one knows at a specific time the sequence $\{b_1, b_2, ..., b_i\}$, then one must not be able to identify the next bit b_{i+1} of the sequence with probability different than $1/2$. Besides

unpredictability of next bit, a random binary sequence should be characterized by ***independency of bits*** (unbiased bits), i.e. each produced bit should not be affected by a previous bit.

Randomness can be evaluated with the application of various statistical tests on a binary sequence. Numerous statistical tests have been proposed to test randomness; a detailed list of statistical tests and suites is provided by Soto [17]. Knuth [20] proposed three statistical tests, named (i) Frequency test, (ii) Run test, and (iii) Combination test. Menezes [14] presented five tests that may provide strong indication of randomness: (i) monobit test, (ii) two-bit test, (iii) poker test, (iv) runs test, and (v) autocorrelation test. Moreover, various test suites have been developed, independently, that perform a variety of statistical tests (battery of tests). The ENT test suite [12] calculates entropy, chi-square metric, arithmetic mean, approximation of pi-value and serial correlation coefficient for any given binary sequence. Marsaglia [28] has implemented another battery of statistical tests, named DIEHARD, which implements more that 15 statistical tests. The Computer Security Division (CSRC[1]) of NIST[2] in accordance with FIPS 140-1 [30] and FIPS 140-2 [23] recommendations, implemented a statistical test suite that performs a variety of tests for evaluation of random number generators [29].

4. DE-SKEWING TECHNIQUES

As mentioned previously, a sequence of bits that is obtained by a source might present (i) *biases,* i.e. the probability of occurrence of 0s is greater than that of 1s or vice versa, and (ii) *correlations,* i.e. the occurrence of bit values depends on previous bits. In order to avoid such phenomena, various techniques have been proposed which, when applied to such sequences, eliminate biases and correlations, producing uniformly distributed sequences.

The first and one of the simplest techniques for de-skewing has been proposed by von Neumann [22]. Considering successive pairs of bits $\{b_i, b_{i+1}\}$, the result of application of von Neumann's function (transition mapping) to such a pair can be seen at the following equations,

$$Neu(\ ): \{0,1\}^2 \rightarrow \{0,1,\Lambda\}$$

$$Neu(b_i, b_{i+1}) = \begin{cases} 0, & \text{if } b_i, b_{i+1} \equiv \{0,1\} \\ 1, & \text{if } b_i, b_{i+1} \equiv \{1,0\} \\ \Lambda, & \text{if } b_i, b_{i+1} \equiv \{0,0\} \text{ or } \{1,1\} \end{cases}$$

where Λ means that nothing is produced as result.

Eastlake et al [34] describe other techniques like parity, Fourier transforms (FFT) and compression that can be used for de-skewing bit sequences.

5. SOURCES OF RANDOMNESS

There exist various sources that can be used to generate true random bit streams. In the following, we describe each source briefly and for each case, we present the source of randomness and how this source can be used for random number generation. Moreover, we refer to the statistical tests that each source is applied to in order to determine the level of randomness.

In general, the scheme to obtain possible random bit streams from a source is shown in Figure 1.

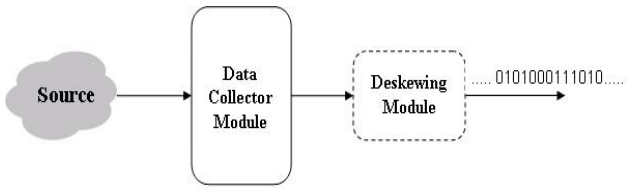

Fig.1 – Obtaining random data from a physical source of randomness. The data collector module collects raw data from the source and feeds them to the deskewing module, which produces possibly unbiased and uncorrelated random bit streams. The de-skewing module is drawn dashed, because its existence is not always necessary.

Basically, the sources that can be used to generate random bit streams with application to embedded systems can be categorized, according to Soohoo [33], as (i) *sampling and amplification of noise sources*, (ii) *dynamical systems*, exploiting the chaotic behavior that small electronic circuits exhibit, and (iii) *oscillator-sampling*.

5.1 Rand 1

Stefanov et al. [8] propose a true random number generator utilizing the intrinsic characteristics of quantum mechanics. In their scheme, photons are generated by an optical LED and are forced to follow two different paths. Upon arrival of each photon, at the end of the path, we can identify which path it has followed and we label one path as '0' and one as '1'. As a result, we get a binary sequence of possibly random bits. The sequence of data has been tested and the source appears to behave like a true

[1] http://csrc.nist.gov/

[2] www.nist.gov/

random source.

5.2 Rand 2

Jakobsson et al. [9] propose a physical random bit generator based on randomness that is exhibited in measurements of access times of various storage devices like hard disks, CD-ROMs, magnetic tapes, etc. Their original work focused on hard disks. They observed that the time needed to read data from a block in a sector of a hard disk cannot be calculated accurately, when we try to access the same block, repeatedly. The response times for repetitive readings of the same block present significant variations and this phenomenon can be used as a source of randomness. Basically, there is evidence that these variations are ought to disk rotational latency caused by air turbulence phenomena that appear into the rotating platters of the hard disk [10].

5.3 Rand 3

Montville [11] conducted research on sources of randomness that can be used in handheld devices, like PDAs, mobile phones, etc. He studied two possible sources: (a) *touch-screen data collection* and (b) *audio data collection*. Considering touchscreen data collection, each point's coordinate in a touch-screen is represented by a 32-bit word, (16 lower bits represent the x coordinate and 16 higher bits represent the y coordinate). The x, y coordinates are XOR-ed between them and the 8 least significant bits of the result are kept. So, each touch-screen point is represented by a byte. Regarding the audio data collection, he used various entropy pools that may produce audio data, like dining in a noisy place, attending a lecture, etc. Those sources are sampled periodically and an 8-bit data sample is kept. In both cases, the random binary sequence is generated by keeping the least significant bit of every produced byte. Then, the raw binary data are de-skewed and submitted to ENT test suite [12], passing successfully the entire statistical test for randomness.

5.4 Rand 4

Fujita et al [15] propose a novel hardware random number generator using as source of randomness the current of a nano-electron channel. They exploit the fact that the current of such a channel can present very big fluctuations, due to the presence of "trap" electrons near the channel, which may drag electrons from the channel making the current fluctuating very quickly. The authors implemented such a nano-device using *a single-electron-transistor (SET) with a single trap*. They derived possibly random binary sequences from the device and passed them from the standard statistical tests for checking randomness that are proposed by NIST [16, 17]. Their results indicate high randomness. Another similar hardware random number generator, which uses as source of randomness the fluctuating current of MOS transistors after soft breakdown has been proposed by Yasuda et al [18]. Generated sequences passed through the same statistical suite, showing high-level of randomness.

5.5 Rand 5

Saitoh et al [19] proposed a physical random number generator based on a variable parametric oscillator (parametron). Considering an unexcited – no RF power excites the circuit - digital parametric circuit, the oscillation phase of the parametron depends on the intrinsic noise that resides into the circuit. Because that noise is statistically random, the oscillation phase presents high fluctuation and cannot be predicted accurately. In their proposed scheme, that unpredictable oscillation phase of such a circuit comprises the source of randomness. The collected random data passed successfully three statistical tests for randomness, specifically (i) Frequency test, (ii) Run test, and (iii) Combination test (details on the test methodologies are provided in [20]).

5.6 Rand 6

Jun and Kocher [21] reviewed and tested the Intel random number generator. The Intel RNG utilizes the thermal noise (Johnson noise), which is present in all resistors as a result of physical and mechanical behavior of materials. The source of randomness comes from two free-running oscillators, one having high frequency and one quite slower. The noise is sampled and amplified and modulates the frequency of the slower oscillator. The fastest oscillator is sampled using the noise-modulated clock, giving random measurements. In turn, those measurements are de-skewed using von Neumann's technique [22] for removing possible biases. The derived binary sequences have been tested for randomness using the DIEHARD test suite, Knuth's tests [20] and FIPS 140-1 recommendation [23], providing strong results. Another similar random number generator has been proposed by Bagini and Bucci [24], who designed a true RNG that can be used for cryptographic purposes, using as source of randomness Gaussian white noise which is amplified and sampled, deriving random binary data. Another commercial true random number generator that uses as source of randomness thermal noise of semiconductors is Random Master [31]; however, it is intended for usage in general-purpose computing

systems.

5.7 Rand 7

Yalcin et al [25] proposed a true RNG using a chaotic oscillator which exhibits a double scroll attractor, as randomness source [26, 27]. The output of the oscillator is a continuous-time chaotic signal and in order to generate sequences of random bits, the authors used two threshold functions, since the state space is sub-divided into three independent regions. Each random bit is generated using the result of the threshold functions as input and finally the produced random sequence is de-skewed for removal of bias and possible bit correlations. The sequences have been extensively statistically tested for randomness using FIPS 140-1 and DIEHARD tests, passing all of them successfully.

Table 1 summarizes the tests under which each source was submitted. For some of the sources, the derived raw binary data have been de-skewed for removal of possible biases and correlations; all schemes have passed successfully all the tests, which they were subjected to.

Table 1 - Testing and Results for each source of randomness

Source	Test	Deskewing	Passed or not
Rand 1	Knuth, entropy, autocorrelation	Yes	Yes
Rand 2	Knuth, DIEHARD	Yes	Yes
Rand 3	ENT	Yes	Yes
Rand 4	Spectral test, NIST	No	Yes
Rand 5	Knuth	No	Yes
Rand 6 [3]	Subset of Knuth, NIST and DIEHARD tests	Yes	Yes
Rand 7	NIST, Diehard	Yes	Yes
Rand 8	Menezes	Yes	Yes
Rand 9	ENT, DIEHARD	No	Yes

[3] We mention the tests that were applied to Intel random number generator, only.

5.8 Rand 8

Nève et al [13] utilize the intrinsic noise of transistors as a source of randomness, developing a System-On-Insulator random signal generator. The noise is used to generate a random signal, which is sampled producing sequences of random bits. These sequences are subjected to de-skewing techniques and they pass successfully the statistical tests for checking randomness, which have been proposed by Menezes [14].

5.9 Rand 9

Rohe [35] proposes a cryptographic random number generator which uses as source of randomness the radioactive decay produced by an ionization-type household smoke detector. The random bit stream is obtained by measuring the time interval between successive decay impulses. Furthermore, he proposes three additional methods to extract randomness from such sources. The obtained random bit streams present uniform distribution (no necessity for de-skewing) and passed successfully the ENT and DIEHARD batteries of statistical tests.

5.10 Comparison

Embedded systems, the systems we focus on, are constrained systems with limited processing capabilities and storage capacity, and most importantly, power limitations. As strong random number generators are necessary for a wide range of applications and services, it is necessary to identify those sources that are effective in embedded systems environments, because several generators have high requirements in terms of processing, power and extra hardware. For example, some sources, such as Rand 3 and Rand 4, take advantage of characteristics of system components, i.e. touch-screen and intrinsic characteristics of transistors, respectively. Thus, such generators can be used in embedded systems, while others are clearly inappropriate, such as Rand 2 that uses accesses to hard disc sectors as the source of randomness.

Table 2 summarizes the application areas of each of the reviewed source of randomness. Clearly, generators appropriate for embedded systems are effective in general-purpose environments as well.

Table 2 – Application area per source

Source	Application Area
Rand 1	General-purpose systems
Rand 2	General-purpose systems
Rand 3	Embedded Systems
Rand 4	Embedded Systems
Rand 5	General-purpose systems & Embedded Systems
Rand 6	General-purpose systems & Embedded Systems
Rand 7	General-purpose systems & Embedded Systems
Rand 8	Embedded Systems
Rand 9	General-purpose systems

5. CONCLUSIONS

Generation of cryptographically strong random numbers is crucial for conventional and emerging applications. It is necessary to identify appropriate sources of randomness that can be used for efficient random number generation.

In this paper, we have identified various sources of randomness, like optical quantum sources, air turbulence in disk accesses, touch-screen and audio data collection using a PDA, noise and intrinsic phenomena of transistors, parametric oscillators, thermal and white noise and chaotic behavior of electronic circuits. Several of these sources appear to provide strong random bit sequences and are appropriate for use in embedded systems that are characterized by constrained resources.

5. REFERENCES

[1] R. von Mises, Grundlagen der Wahrscheinlicheitsrechnung, Math. *Z.,* pp. 52-99, 1919.

[2] Compagner, Definition of Randomness, *Amer. J. Phys.,* vol. 59, pp. 700-705, 1991.

[3] S. Wolfram, *A New Kind of Science,* Wolfram Media, 2002.

[4] G.J. Chaitin, Randomness and Mathematical Proof, *Sci. Am.,* vol. 232, pp. 47, 1975.

[5] G. Chaitin, Information-theoretic computation complexity, *IEEE Trans. on Inf. Theory,* vol. 20, pp. 10-15, 1974.

[6] C.E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.,* vol. 27, pp. 379-423, July 1948. 1948.

[7] C.E. Shannon, Communication theory of secrecy systems, *Bell Sys. Tech. J.,* vol. 28, pp. 656-715, 1949. 1949.

[8] Stefanov, N. Gisin, O. Guinnard, L. Guinnard and H. Zbinden, Optical Quantum Random Number Generator, available at *http://arxiv.org/abs/quant-ph/9907006,* 1999-07-02. 1999.

[9] M. Jakobsson, E. Shriver, B.K. Hillyer and A. Juels, A practical secure physical random bit generator, *in Proceedings of the 5th ACM conference on Computer and communications security*, 1998, pp. 103-111.

[10] D. Davis, R. Ihaka and P. Fenstermacher, Cryptographic Randomness from Air Turbulence in Disk Drives, *in Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology*, 1994, pp. 114-120.

[11] A.W. Montville, Random Number Generation on Handheld Devices for Cryptographic Applications. *M.Sc. Thesis*, Oregon State University, May 21, 2003.

[12] J. Walker, ENT – A Pseudo Random Number Sequence Test Program, *http://www.Fourmilab.ch/random/,* 2003.

[13] Nève, D. Flandre and J. Quisquater, Feasibility of Smart Cards in Silicon-On-Insulator (SOI) Technology, *in USENIX Workshop on Smartcard Technology*, 1999, pp. 1-9.

[14] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, Pseudorandom Bits and Sequences, chapter in *Handbook of Applied Cryptography,* 5th ed., CRC Press, 2001, pp. 169-190.

[15] S. Fujita, K. Uchida, S. Yasuda, R. Ohba and T. Tanamoto, Novel Random Number Generators Based on Si Nanodevices for Mobile Communication Security Systems, in *Technical Proceedings of the 2003 Nanotechnology Conference and Trade Show*, 2003, pp. 309-312.

[16] Rukhin, J. Soto, J. Nechvatal, M. Smid and E. Barker, A Statistical Test Suite for Random and Pseudo-Random Number Generators for Cryptographic Applications, *NIST Special Publication 800-22,* May 15, 2001.

[17] J. Soto, Statistical Testing of Random Number Generators, in *Proceedings of the 22nd National Information Systems Security Conference*, 1999.

[18] S. Yasuda, H. Satake, T. Tanamoto, R. Ohba, K. Uchida and S. Fujita, Physical random number generator based on MOS structure after soft breakdown, *IEEE Journal of Solid-State Circuits,* vol. 39, pp. 1375-1377, Aug., 2004.

[19] S. Yoshiaki, H. Junichi, N. Hiroshi and K. Tohru, Generation of physical random numbers with a variable-capacitor parametron, *Electronics and Communications in Japan (Part III: Fundamental Electronic Science,* vol. 86, pp. 24-32, 31 Oct 2002. 2003.

[20] D.E. Knuth, *The art of computer programming, volume 2 (3rd ed.): Seminumerical algorithms,* Addison-Wesley Longman Publishing Co., Inc, 1997.

[21] B. Jun and P. Kocher, The Intel Random Number Generator, Cryptography Research Inc., *White Paper,* 22-4-1999.

[22] J. von Neumann , Various techniques for use in connection with random digits, in *John von Neumann's Collected Works,* vol. 5, A.H. Taub Ed. Pergamon Press, 1963, pp. 768-770.

[23] FIPS 140-2, NIST, 2002, available at http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[24] V. Bagini and M.E. Bucci, A Design of Reliable True Random Number Generator for Cryptographic Applications, *in Proc. of Cryptographic Hardware and Embedded Systems (CHES),* LNCS 1717, pp.204-219, 1999.

[25] M.E. Yalcin, J.A.K. Suykens and J. Vandewalle, True Random Bit Generation from a Double Scroll Attractor, *Tech. Rep.,* ESAT-SISTA, K.U.Leuven., Belgium,. Internal Report 03-84, 2003.

[26] L. Chua, M. Komuro and T. Matsumoto, The double scroll family, *IEEE Transactions on Circuits and Systems,* vol. 33, pp. 1072-1118, 1986.

[27] L.O. Chua, C.W. Wu, A. Huang and G. Zhong, A universal circuit for studying and generating chaos. II. Strange attractors, IEEE *Transactions on Circuits and Systems I: Fundamental Theory and Applications,* vol. 40, pp. 745-761, 1993.

[28] Marsaglia, G. DIEHARD Test Suite, http://stat.fsu.edu/pub/diehard/, Florida State University, 1997.

[29] NIST Statistical Test Suite for Testing Randomness, http://csrc.nist.gov/rng/ .

[30] FIPS 140-1, Security Requirements for Cryptographic Modules, NIST, 1994, available at http://www.itl.nist.gov/fipspubs/fip140-1.htm.

[31] Random Master [Online] http://www.t-rs.co.jp/eng/products/index.htm

[32] Ian Goldberg, David Wagner, Randomness and the Netscape Browser, *Dr. Dobb's Journal*, online at http://www.ddj.com/documents/s=965/ddj9601h/

[33] Soohoo, A., Lockdown! Random Numbers Secure Network SoC Designs, *Communication Systems Design*, 2003, online at http://www.commsdesign.com

[34] D.E. Eastlake, J.I Schiller, S. Crocker, RFC 1750 - Randomness Requirements for Security, *RFC Archives*, 1994, Network Working Group, online at http://www.faqs.org/rfcs/rfc1750.html

[35] Rohe, Markus, RANDy – A True Random Number Generator based on Radioactive Decay, online at http://www-krypt.cs.uni-sb.de/projects/randy/randy.pdf

A. Fragopoulos received his BSc in Physics in 2000 from University of Patras. Since 2002, he is a PhD student at the Department of Electrical and Computer Engineering, University of Patras. His current research involves embedded systems and security, random numbers generation and cryptography. He is a student member of IEEE and USENIX, a member of IEEE Computer Society, and a member of ACM.

D. Serpanos is a Professor at the Department of Electrical and Computer Engineering, University of Patras. He holds a PhD in Computer Science from Princeton University (1990) and a Diploma in Computer Engineering and Information Sciences from the University of Patras (1985). Prof. Serpanos was a Research Staff Member at IBM Research, T.J. Watson Research Center (1990-1996) and a faculty member of Computer Science at the Univ. of Crete, Greece (1996-2000). His research interests include computer architecture, network systems architecture, security systems and multimedia systems. He is a Senior Member of the IEEE, a member of ACM, NYAS and the Technical Chamber of Greece, and an educational member of USENIX.