Random ambience using high fidelity images

Nur Azman Abu and Shahrin Sahib

Faculty of Information and Communication Technology,
Universiti Teknikal Malaysia Melaka (UTeM),
Melaka, Malaysia

ABSTRACT

Most of the secure communication nowadays mandates true random keys as an input. These operations are mostly designed and taken care of by the developers of the cryptosystem. Due to the nature of confidential crypto development today, pseudorandom keys are typically designed and still preferred by the developers of the cryptosystem. However, these pseudorandom keys are predictable, periodic and repeatable, hence they carry minimal entropy. True random keys are believed to be generated only via hardware random number generators. Careful statistical analysis is still required to have any confidence the process and apparatus generates numbers that are sufficiently random to suit the cryptographic use. In this underlying research, each moment in life is considered unique in itself. The random key is unique for the given moment generated by the user whenever he or she needs the random keys in practical secure communication. An ambience of high fidelity digital image shall be tested for its randomness according to the NIST Statistical Test Suite. Recommendation on generating a simple 4 megabits per second random cryptographic keys live shall be reported.

Random, ambience, live key generation, true random number generator, high fidelity image.

1. INTRODUCTION

The security of modern cryptographic system should be no longer based on the secrecy of the algorithm system design but rather on the randomness of the key being used. There are mainly two separate ways for generating random keys. First, the random bit can be captured from random phenomena by using a physical device. This strategy takes various factors, such as noise and time of day, into account. Extremely complex hardware random number generators are based on essentially random atomic phenomena such as quantum physics, radioactive decay and thermal noise.

Second, random keys can also be generated computationally, by using algorithms. These so called pseudorandom keys are sequences of numbers which approximate many of the properties of random keys. Pseudorandom keys can easily be regenerated again and thus are not truly random in nature. Nevertheless, they are still preferred especially on large input or plaintext which requires long keys. Furthermore, most of the cryptographic operations nowadays mandates random key as an input. These operations are mostly designed and taken care of by the developers of the cryptosystem. The security of such a one-time key cryptosystem, for instance, relies heavily on the design and the trustworthiness of the developer oneself.

In this paper, a new technique is proposed using readily available apparatus which is capable of generating large random key live on demand. The technique has been designed based on the air ambience principle. The rest of the paper shall be organized as follows. Section 2 will give an overview of physical random number generators and their current challenge. Section 3 will discuss the importance of true random keys in cryptosystem. Section 4 shall introduce the concept of air ambience as a feasible source of randomness. In section 5 gives the challenges to meet on which strict requirements have been imposed on the random key generation apparatus. Section 6 discusses on the selected random statistical tests for one-mega bit key. Section 7 gives a sample experimental result. Section 8 concludes the paper.

2. PHYSICAL RANDOM NUMBER GENERATOR

Cryptography still remains an important science in today's civilization, be it for commercial use or the privacy of individuals. The emergence of electronic commerce and multimedia network has made it a necessity to use cryptosystem nowadays. Every cryptographic operation nowadays mandates random key as an input. In principle, true random number generators must be able to capture randomness from physical phenomena. The physical phenomenon can be very simple such as the little variations in user's mouse movements or in the time difference between keystrokes from the user's typing speed. These techniques, though friendly, produce limited random bits.

Third International Conference on Digital Image Processing (ICDIP 2011),
edited by Ting Zhang, Proc. of SPIE Vol. 8009, 80092H · © 2011 SPIE
CCC code: 0277-786X/11/$18 · doi: 10.1117/12.896308

Due to the nature of random numbers, pseudorandom number generators are preferred instead due to their efficiency and reliability.[1] As the name suggests, pseudorandom numbers are not truly random. Rather, they are generated from a mathematical formula. The outputs of pseudorandom number generators may exhibit most of the theoretical properties of random numbers. However, pseudorandom numbers are predictable, periodic and repeatable by the developer of the cryptosystem. The use of pseudo-random processes to generate secret quantities can result in pseudo-security.[2] The use of pseudo random number generator is quite misleading. It still remains a common problem today. It can be observed in the latest textbook on how to generate a simple random number in iPhone application development.[3] The author suggests on how to generate random number via the random function from a regular old C function by setting the seed to the system clock. System clock is a 32-bit number. It carries only few bits entropy.

This vulnerability appears also in script programming. PHP is a widely-used general-purpose scripting language. Since version 4.2.0 PHP automatically seeds the random number generators on the first usage of Mersenne Twister pseudo random number generator.[4] Unfortunately it was discovered[5] that macro script to generate seed contains several problems that can lead to a weaker seed than expected. In the worst case, the seed is directly predictable, which allows the solution provider to predict all pseudo random numbers being generated. Even recently similar problem still appears. Usage of weak random number generation in password reset functionality allows predicting the password reset token and the randomly generated password, which results in account takeover.[6] The problem originates from the uses of Mersenne Twister pseudo random number generator to generate the random strings and its state is known to a remote attacker. He is therefore able to predict both the generated password reset token and also the new password. This flaw allows a hacker to take over arbitrary accounts.

3. RANDOM KEY IN CRYPTOSYSTEM

True random numbers are believed to be generated only using hardware random number generators. Careful statistical analysis is still required to have any confidence the process and apparatus generates numbers that are sufficiently random to suit the cryptographic use. The security of the open cryptosystem must reside only in the key being used. Now, the key is the main formula, or answer, to lock and unlock the encryption. It cannot be just any number as a key. It must be random key. In this underlying principle, each moment in life is considered unique in itself. The random key must be unique for the given moment generated by the user whenever he or she needs the random bits in practical cryptographic applications.

There are many random keys in cryptographic system. Every operation in cryptosystem requires random key. Typically, Session Key is the most critical which should be regenerated on every operation. It is commonly used as secret key for symmetric block cipher. Since it is not very easy to generate true random key every time during encryption process, a pseudo random number generator shall be used. However, a pseudo random number generator still requires a random seed to initialize the generation process. Otherwise, the developer of the cryptosystem shall gain full control of the cryptosystem. In a modern cryptosystem such as elliptic curve cryptosystem (ECC), every encryption process will also require another random Encrypt Key. Although it is not well known, the mandatory use of Encrypt Key in a popular encryption mode in ECC has been well written in Cryptography textbook. [7]

Unlike the Session Key, this Encrypt Key shall be the secret key for as-symmetric block cipher or public key infrastructure (PKI). Based on the Digital Signature Algorithm,[8] every time a user would like to produce Digital Signature using his or her PKI of certain message for authentication requires random key besides his or her own private key. The DSS signature algorithm requires the cryptosystem to generate a new random number with every signature.[9] In key distribution and reciprocal authentication schemes, two communicating parties cooperate by exchanging message to distribute keys and authenticate each other. The use of random nonce frustrates a third party to determine or guess the nonce.[10]

Ultimately, the keys are stored in the system and the entire system may depend on a single master key which must be random. This single master key may be exposed, lost or destroyed. Having copies of the same the supreme master key to more people will increase the vulnerability of the system from betrayal. Then, comes threshold scheme[11] which provides a solution by breaking the master key into several shadow keys to highest board of panels. This shadows key operates based on majority presence of the shadow keys would be equivalent to having master key. Still, threshold scheme also needs another random key to produce shadow keys from a single master key.

4. RANDOM AMBIENCE

Physical hardware random number generator has a greater advantage, since it can produce completely unpredictable random sequences. Most of the true random number generators recently are designed based on special hardware devices such as quantum detector devices,[12] electronic flip-flop[13] and/or chaos oscillator digital circuits.[14] In this research study, however, the true random number generator does not depend on a special device. Essentially, it is the environment which becomes the source of randomness. Air ambience is the natural choice here. The movements of microscopic particles are random. They are randomly suspended in the air caused by collisions among particles and the gas molecules of air. Even though air appears thin, it is in fact a heavily dense environment. Density of air is about 1.29 kg m^{-3} or 0.0129 g cm^{-3}. Average mass of air (a mixture of different gases) is 28.9 g/mol. as shown in Table 1 One mole is 6.02 x 10^{23} particles, namely, Avogadro's number.

Table 1. The molecular mass of major air molecules.

Molecule	Volume Ratio in Air	Molar Mass (g/mol)	Molecular Mass in Air
Oxygen	0.2095	32.00	6.704
Nitrogen	0.7809	28.02	21.880
Carbon Dioxide	0.0003	44.01	0.013
Hydrogen	0.0000005	2.02	0
Argon	0.00933	39.94	0.373
Total Molar Mass of Air			28.970

Apparently, air molecules move in random directions, at high speed. Average velocity of a single air particle is around 500 m/s at room temperature 27°C or 300 Kelvin. For oxygen O_2, the mass M = 36 g/mol at the room temperature of 298 Kelvin = 25°C, its velocity is approximately 444 m/s. Assuming the molecular collisions driving the motion are completely random, the motions in the three directions are uncorrelated. Based on the basic principles of physics, the molecules in the atmosphere obey Newton's law of motion. Nevertheless, as a whole they move randomly, meaning, any molecule may move in any direction with any speed. At any given moment, a certain percentage of molecules move at high speeds and a certain percentage move at low speeds.[15] The movement of these molecules in the air generates the random ambience.

5. THE CHALLENGES IN GENERATING CRYPTOGRAPHIC KEYS

Many physical generators can only generate short keys, live on demand, of insufficient length for modern cryptographic protocols. Others requires expensive special device such as radioactive scanner. The users and owners of cryptosystem are mostly nontechnical. It is important to have a generation process and apparatus which are physical, economics, convenient, efficient to use and secure. The system apparatus should not only just use only requires minimum hardware and but also utilize only few basic computer algorithms. The randomness should come from the source rather than the algorithm itself. The measuring device should work automatically on demand from user's physical environment. In order to make sure the randomness of the numbers being generated, they should pass a full set of statistical tests once generated.

Practically, the keys are to be generated on demand from user's physical environment or somehow related to the user. In a way, the apparatus must be mobile which can be carried around without much hassle. At the same time, it must be robust enough to produce random bits consistently under various climates. The underlying principle of this research, each moment in life is considered unique in itself. The random key is unique for the given moment generated by the user whenever he or she needs the random bits in a practical cryptographic application.[16]

6. ONE MEGABIT RANDOM TESTS

The live experiment in this research project requires dedicated random statistical tests for large input. The candidate of random key here is a block of 2^{20} = 1,048,576 bits. A statistical test is formulated to test a specific null hypothesis (H_0).

For the purpose of this statistical test, the null hypothesis under test is that the binary sequence $\boldsymbol{\varepsilon}$ being tested is random against the alternative hypothesis (H_1) for which the binary sequence $\boldsymbol{\varepsilon}$ is not random.

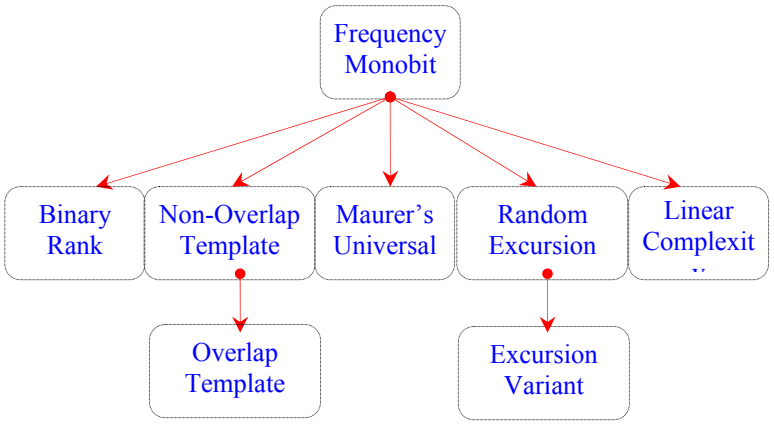

Figure 1. The hierarchy of large random tests : Failing higher test will ensure the failure of the lower test.

For each statistical test, a set of *p*-values (corresponding to the set of sequences) is produced. For a fixed significance level, a certain percentage of *p*-values are expected to indicate failure. For example, if the significance level is chosen to be 0.01 (i.e., $\alpha = 0.01$), then about 1% of the sequences are expected to fail. A sequence passes a statistical test whenever the *p*-value $\geq \alpha$ and fails otherwise. The parameter α denotes the significance level that determines the critical region of acceptance and rejection. Even though NIST recommends that α be in the range [0.001, 0.01], for this large cryptographic keys it is still practical to use smaller significant level α such as 0.001. Only 8 tests are particularly suitable for large cryptographic keys size. The selected random tests from NIST[17] for one megabit key are listed in the Table 2 below.

Table 2. The list of suitable random tests for large one megabit numbers.

index	Statistical Test
1	Frequency Monobit Test
2	Binary Matrix Rank Test
3	Non-overlapping Template Matching Test
4	Overlapping Template Matching Test
5	Maurer's Universal Test
6	Random Excursions Test for positive states $x = +1, +2, +3, +4$
7	Random Excursions Test for negative states $x = -1, -2, -3, -4$
8	Random Excursions Variant Test for positive state $x = +1, +2, \ldots, +9$
9	Random Excursions Variant Test for negative state $x = -1, -2, \ldots, -9$
10	Linear Complexity Test

The 8 selected tests including the first Frequency Monobit Test are basically relies heavily on the randomness of the binary sequence. Figure 1 shows the hierarchy of the tests. Once a particular block key set fails one test it is considered non-random and will certainly fail the next test in lower hierarchy. All 8 random tests have been coded using 32-bit processing style in order to achieve real time performance.[18] The *p*-value indices are shown in Table 2 for easy reference.

7. EXPERIMENTAL RESULTS

In this research, the randomness of air ambience is tested to its full potential. The objective is to generate one megabit per snapshot. A typical 14-bit medium range digital camera has been selected as the source of random input. The digital

camera usually comes with its own software package driver. In this experiment Canon EOS 50D has been used. It is crucial the compression mode to be switch off or disable for this particular use. Each pixel should contribute only one bit, namely, the least significant bit. The idea is to capture the air ambience of the moment in life as a unique source of random ambience. The least significant bit of the center square of a digital image is then converted into a one dimensional binary sequence via circular reading. The circular scan should start from the center of the image as proposed in previous work.[19] The long binary sequence shall been tested using the selected 8 NIST Statistical Tests[18] suitable only for long binary sequence.

Figure 2. A sequence of 4 frames captured in continuous snapshots and random ambience is visualized on the center image.

In order to produce several megabit output, a sequence of images shall be captured using in continuous snapshots. This particular model is capable of capturing 5.6 frames per second. Even though each frame consist of more than 5 mega pixels, preferably, the center one megapixels only shall be taken via circular scan. Samples images of 4 frames are shown in Figure 2 together with the least significant bit of the 1024 by 1024 bits in center image appears to be noise bits. The binary sequence intended cryptographic keys shall then be tested for their randomness using selected random tests suitable for large number.

Table 3. The 32 *p*-values of 4 one-megabit random tests on 4 continuous frames

Test Index	Frame 1	Frame 2	Frame 3	Frame 4	Test Index	Frame 1	Frame 2	Frame 3	Frame 4
1	0.16257	0.50540	0.50166	0.83599	8(x = +4)	0.24645	0.34075	0.25038	0.48337
2	0.60664	0.32671	0.73176	0.91971	8(x = +5)	0.37719	0.72418	0.41100	0.46381
3	0.44954	0.74102	0.03204	0.01585	8(x = +6)	0.60603	0.98181	0.54901	0.72481
4	0.38233	0.06060	0.99685	0.33564	8(x = +7)	0.83501	0.93870	0.74486	0.99620
5	0.72050	0.36514	0.49995	0.23226	8(x = +8)	0.95275	0.90674	0.85285	0.65432
6(x = +1)	0.66886	0.36511	0.12057	0.41602	8(x = +9)	0.63077	0.67763	0.97683	0.48157
6(x = +2)	0.79333	0.35768	0.38745	0.21395	8(AveP)	0.64229	0.60210	0.45213	0.48338
6(x = +3)	0.65584	0.12722	0.05047	0.33917	9(x = −1)	0.13311	0.57922	0.47253	0.46032
6(x = +4)	0.61051	0.40668	0.25271	0.34677	9(x = −2)	0.11463	0.33682	0.55219	0.88962
6(AveP)	0.68214	0.31417	0.20280	0.32898	9(x = −3)	0.16452	0.18348	0.29892	0.89007

7($x = -1$)	0.08320	0.91056	0.21357	0.61497	9($x = -4$)	0.14037	0.08638	0.08226	0.73088
7($x = -2$)	0.16073	0.67756	0.24804	0.37505	9($x = -5$)	0.11770	0.14148	0.13765	0.94069
7($x = -3$)	0.25522	0.53526	0.44146	0.27438	9($x = -6$)	0.19509	0.32317	0.43130	0.72481
7($x = -4$)	0.20820	0.20820	0.45487	0.16137	9($x = -7$)	0.37293	0.40153	0.73483	0.92791
7(AveN)	0.17684	0.58289	0.33948	0.35644	9($x = -8$)	0.51806	0.33545	0.59492	0.67364
8($x = +1$)	0.72287	0.14376	0.11402	0.05448	9($x = -9$)	0.56752	0.38201	0.59492	0.62021
8($x = +2$)	0.91368	0.29474	0.06199	0.09782	9(AveN)	0.25822	0.30773	0.43328	0.76202
8($x = +3$)	0.49586	0.41058	0.10820	0.39403	10	0.91686	0.10033	0.04694	0.70661

Even though the pictures seem to appear similar, according to the ambience principle, they must independent from one another in term of their least significant bits. The sequence of 4 frames shall be tested according to one megabit random statistical test per frame. The results are displayed in Table 3. The 32 *p*-values for the first frame are visualized shown in Figure 3. They can easily pass all the random statistical tests at significant level $\alpha = 0.01$.

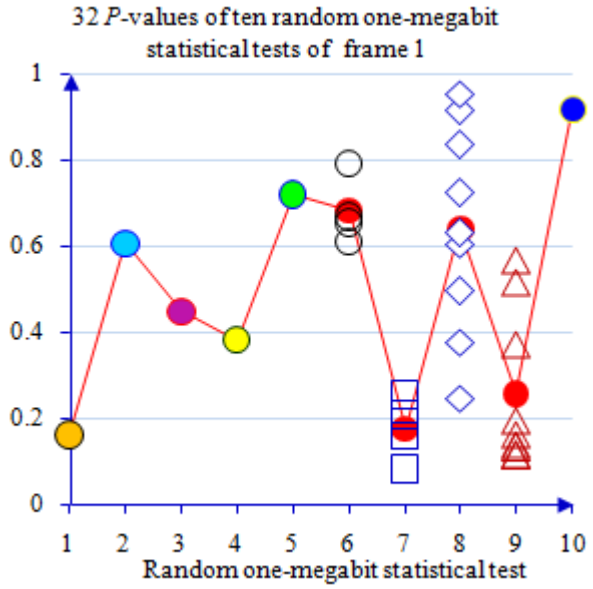

Figure 3. The 32 *p*-values one-megabit random tests on 4 continuous snapshots for frame 1 are plotted with their average values.

8. CONCLUSIONS

The aim of this paper is to show the usage of air ambience for generating true random numbers. Due to difficulty of proving unpredictability in a theoretical way, the proposed true random live output bit is subjected to set of statistical tests. Due to the air ambience, the collections of least significant bits in the high fidelity digital image have shown to easily pass selected statistical tests. Current digital camera is capable of capturing high quality image. Thus, image from user's environment can be a good source of random cryptographic key. In this paper, the least significant bit of the digital image has been statistically proven to be random. In certain practical cryptographic application it is crucial to produce large cryptographic key live on demand. This research provides a practical avenue to generate large random key within short period of time up to 4 megabit per second.

This research project has been designed to support practical cryptographic operations in the near future. A step-by-step procedure has been developed to produce large 2^{20} bit keys per snapshot. Selected NIST random tests have been used to check on the binary sequence. The one megabit cryptographic key has been generated per digital image and test for randomness. All 36 *p*-values out of random statistical tests are larger than $\alpha = 0.01$. Air ambience is a good source for generating true random megabits.

9. FUTURE RESEARCH

Further research is still needed to explore the principle or air ambience as a source of randomness. A special dedicated hardware and software well equipped with microphone and camera may be designed to automatically generate the random ambience key. The system should take care of the warning mechanism in the case of producing non-random keys. A special dedicated photo sensor may be designed to automatically generate several megabit random number per snapshot. The random statistical tests may be directly embedded on board. A further research study should also be done on the extreme environmental condition where the ambience may not produce random numbers. A dedicated photo sensor capable of capturing 16-bit per pixel per color should be looked into. The current medium range digital camera has shown to be capable of capturing 14-bit per signal sample for one megabit random generation process. Finally, it is worthwhile to investigate the depth of bit per signal sample which would better produce random bit in future research.

10. ACKNOWLEDGMENT

The authors would like to express sincere appreciation to Universiti Teknikal Malaysia Melaka for giving full technical and financial supports in this research study.

11. REFERENCES

[1] Barker, E. and Kelsey, J., "Recommendation for Random Number Generation Using Deterministic Random Bit Generators," NIST Special Publication 800-90(Revised), (2007).

[2] Eastlake, D. 3rd, Schiller, J. and Crocker, S., Randomness Requirements for Security, BCP 106, RFC 4086, (2005).

[3] Drake, M. J., "How to Use Random Numbers in Your iPhone App," Learn How to develop an iPhone Apps, 6 May 2009.

[4] Matsumoto, M. and Nishimura, T., "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," ACM Transaction on Modeling and Computer Simulation 8(1), 03-30 (1998).

[5] Esser, S., "PHP GENERATE_SEED() Weak Random Number Seed Vulnerability," Security Advisory, 6 May 2008.

[6] Stefan Esser, "MyBB Password Reset Weak Random Numbers Vulnerability," Security Advisory, 13 April 2010.

[7] Stallings, W., [Cryptography and Network Security: Principle and Practice], Prentice Hall, 3rd Edition, 306-307 (2004).

[8] Digital Signature Standard, Federal Information Processing Standards Publication 186-2, (2000).

[9] Bellare, M., Goldwasser, S. and Micciancio, D., "Pseudo-Random Number Generation within Cryptographic Algorithms: the DSS Case," Proceedings Advances in Cryptology-Crypto 97, Lecture Notes in Computer Science 1294, Springer-Verlag, (1997).

[10] Stallings, W., [Cryptography and Network Security: Principles and Practice], Prentice Hall, 5th Edition, 219-220 (2010).

[11] Desmedt, Y. and Frankel, Y., "Threshold Cryptosystems," In Proceeding of CRYPTO '89, 307-315 (1989).

[12] Fürst, M., Weier, H., Nauerth, S., Marangon, D. G., Kurtsiefer, C. and Weinfurter, H., "High Speed Optical Quantum Random Number Generation," Optics Express 18(12), 13029-13037 (2010).

[13] Thamrin, N. M., Witjaksono, G., Nuruddin, A. and Abdullah, M. S., "A Photonic-based Random Number Generator for Cryptographic Application," 9th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 356-361 (2008).

[14] Danger, J.-L., Guilley, S. and Hoogvorst, P., "High Speed True Random Number Generator based on Open Loop Structures in FPGAs," Microelectronics Journal 40(11), 1650-1656 (2009).

[15] Serway, R. A. and Jewett, J. W., [Principles of Physics], 3rd Edition, Brooks Cole, 564−569 (2003).

[16] Abu, N. A. and Sahib, S., "Random Ambience Key Generation Live on Demand," Proceedings 2nd International Conference on Signal Processing Systems (ICSPS 2010), Volume 1, 110-114 (2010).

[17] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J. and Vo, S., "A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications," NIST Special Publication 800-22, Revision 1, August (2008).

[18] Abu, N. A. and Sahib, S., "One Megabit Random Ambience," International Journal of Cryptology Research 2(1), 073-087 (2010).

[19] Lang, W. S., Abu, N. A. and Sahib, S., "Cryptographic Key from Webcam Image," International Journal Cryptology Research 1(1), 115−127 (2009).