International Journal of Cryptology Research 1 (1): 115-127 (2009)

Cryptographic Key From Webcam Image

Wong Siaw Lang, Nur Azman Abu, Shahrin Sahib
Faculty of ICT, Universiti Teknikal Malaysia Melaka
Email:nura@utem.edu.my

ABSTRACT

Random numbers plays a critical role in cryptosystem. Random key is recognized as an essential part of the security of cryptographic system. Even more today, it has become as an integral part of cryptographic algorithms. Due to the nature of random numbers, pseudorandom numbers are preferred instead due to it efficiency and reliability. 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 some of the properties of random numbers. Pseudorandom numbers are predictable, periodic and repeatable by the developer of the cryptosystem. Truly 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. In this paper, user's image from digital webcam shall be tested for its randomness according to the NIST Statistical Test Suite. Recommendation on using webcam images as source of random cryptographic keys shall be reported.

OVERVIEW

Design Criteria in Modern Cryptographic Era are as follows:

i. The algorithm must provide a high level of security.
ii. The algorithm must be completely specified.
iii. The algorithm must be clear easy to understand by cryptographic community.
iv. The security of the algorithm must reside in the key
v. The security should not depend on the secrecy of the algorithm.
vi. The algorithm shall be made public and available to all users.
vii. The algorithm must be adaptable for use in diverse future applications.
viii. The algorithm must be economically implementable in electronic devices.
ix. The algorithm must be efficient to use.
x. The algorithm must be validated.
xi. The algorithm must be exportable.
xii. The basic algorithm must have been tested and certified.

The security of modern cryptographic system should 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 numbers. 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 radioactive decay and thermal noise. Second, random numbers can also be generated computationally, by using algorithms. The second method generates so called pseudorandom numbers which are sequences of numbers which approximate many of the properties of random numbers. Pseudorandom numbers can easily be generated again and thus are not truly random in nature, like the hardware generated ones[1].

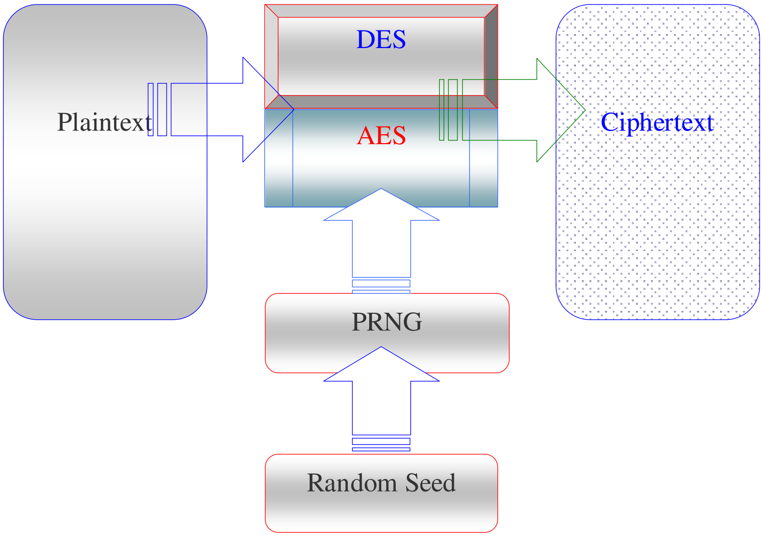

Figure 1: Pseudo-random Number Generator requires random seed in order to be secure

This study on generation of random numbers is intended to support practical cryptographic operations. A step-by-step procedure shall be developed to produce short 128, 256, 512 and 1024-bit keys. The rest of the paper shall be organized as follows. In Section 2, previous study on several physical random number generators shall be describes briefly including the criteria of the random number generating process and apparatus. Section 3 turns on the importance of an user's physical environment. Section 4 shows few places in cryptographic operation which requires random keys. Section 5 will go through the user's webcam images as a source of random key. Section 6 explains on the selected random tests being used in this study and Section 7 will give the experimental result and conclusion is made in Section 8.

PHYSICAL RANDOM NUMBER GENERATOR

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. There are several projects in using physical phenomena to generate random numbers. RANDOM.ORG is a true random number service that generates randomness via atmospheric noise captured using a normal radio[2].

A more complex physical phenomenon to capture randomness is from radioactive source. The HotBits service at Fourmilab in Switzerland uses this technique[3]. The points in time at which a radioactive source decays are strongly believed to be unpredictable. The random source is fundamentally governed by the inherent uncertainty in the quantum mechanical laws of nature. The random bits, nicknamed HotBits, are generated by timing successive pairs of radioactive decays detected by a Geiger-Müller tube interfaced to a computer. HotBits generation hardware produces data at a modest rate 100 bytes per second,

Another moderate project is LavaRnD. The idea is to capture noise from a charge-coupled device(CCD). The typical webcam CCD is enclosed in a light-proof container, and operated at a high gain. The resulting digital snapshots which are not perfectly black contain noise. The LavaRnd system takes noisy data from the CCD and runs it through an algorithm called the Digital Blender, combination of different SHA-1 cryptographic hash operations running in parallel, and different xor-rotate and fold operations[4].

Physical hardware random number generator has a greater advantage, since it can produce completely unpredictable random sequences. Pseudorandom number generators are usually combined with a physical generator to obtain both the speed and unpredictability. The physically hardware generated numbers shall be used as a seed, an initial number value, and the software implemented algorithm produces random sequences from that seed as shown in Figure 1 above. The seed is supposed to be truly random and it is therefore important to be careful when choosing or generating it, given the seed and parameters of the pseudo-random number sequence, one can reproduce the exact same appeared-to-be random sequence.

The criteria of the generating process and apparatus should be as follows:

i. Physical source
ii. Minimum hardware requirements
iii. Economics
iv. Minimum mathematical formula

v. Involve only basic computer algorithm
vi. Pass Statistical tests
vii. Convenient to users
viii. Efficient
ix. Measuring device should work automatically
x. Generated on demand from user's physical environment

Most techniques can only generate short keys that are of insufficient length for modern cryptographic protocols. Others requires expensive special device such as radioactive scanner. The last criteria requires the keys to be generated on demand from user's physical environment or somehow related to the user.

USER'S PHYSICAL ENVIRONMENT

There are several studies have been done on capturing random numbers on demand from user's physical environment which satisfies the last criteria. In another study, a technique had been proposed to generate a biometric key from a user's voice while speaking a password[5]. In recent years researchers have turned towards merging biometrics with cryptography as a means to improve overall security. A biometric based binary sequence (bio-key) can be generated from 3D face images[6]. Previously, the second author has studied on raw white room noise a source of cryptographic keys[7]. In this study, the key shall be generated based on typically user's image from digital webcam. The key is not really coming from the user facial image rather it is coming the user's room ambience shall be taken into consideration.

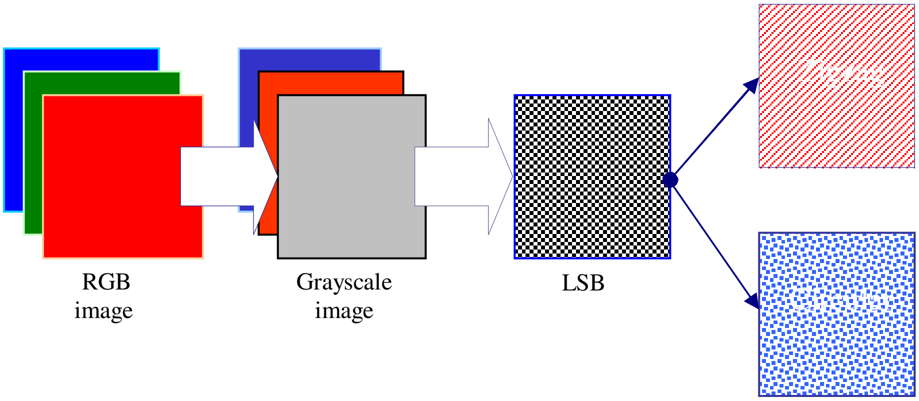

Figure 2a: Color image is converted to grayscale image before taking LSB into zigzag or circular reading.

In this underlying study, 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 practical cryptographic applications.

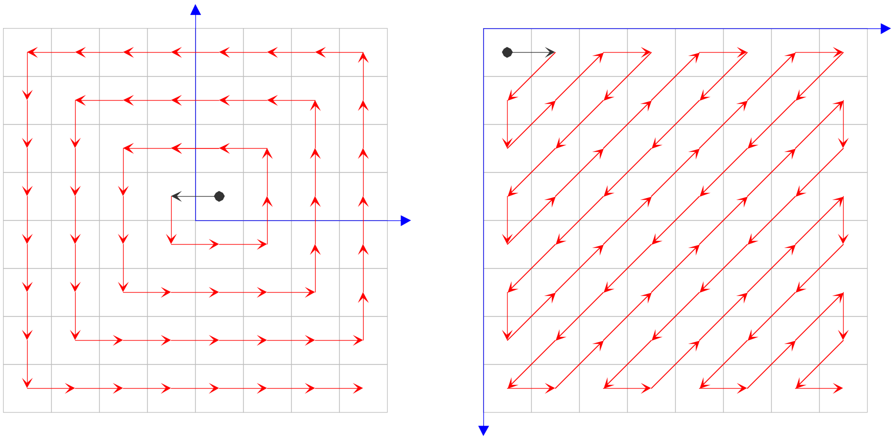

Figure 2b: The circular scan on the left in this case is preferred to zigzag scan on the right

RANDOM KEYS IN CRYPTOSYSTEM

The need for random and pseudorandom numbers arises in many cryptographic applications as the following[8]:

i. Session key for Symmetric Encryption
ii. Private key for Public Key Infrastructure
iii. Encryption Key
iv. Random numbers in digital signing
v. Master key for cryptosystem
vi. Random numbers for generating shadow keys in Threshold Scheme

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 main reason behind idea of this study is to check on the validity of having white room noise as a source of random keys. A developer of stream cipher system can easily made a claim that his/her system is highly secure since the system use room noise to capture the random key. The security of such an OTK (One-Time Key) cryptosystem relies heavily on the design and the trustworthiness of the developer oneself.

WEBCAM IMAGES AS SOURCE OF RANDOM INPUT

In this study, a typical web camera on personal computer has been selected as the source of random input. A webcam image consists of 240 by 320 raw color pixels. The webcam usually comes with its own software package driver. It is crucial the webcam compression 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 ambience noise of the user in his or her own room.

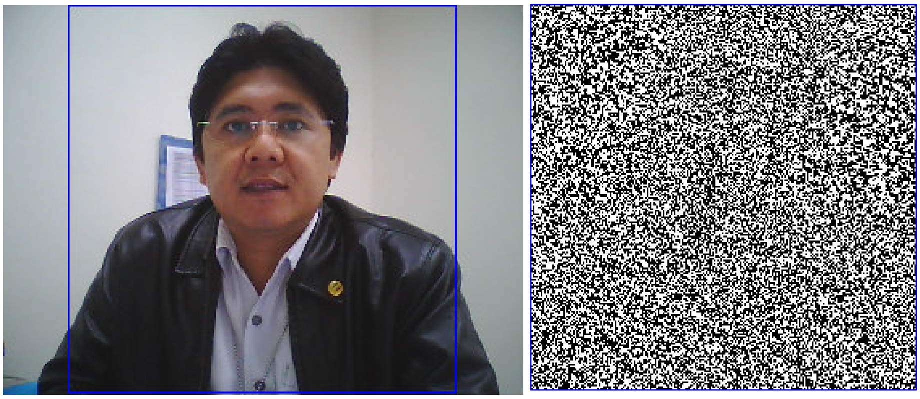

Figure 3: The least significant bit of a sample webcam image on the left is visualized as a plain noise on the right.

The raw RGB image is first has to go through the affine transformation in colour space, converted RGB to YCrCb Conversion. RGB shall be separated into a luminance part (Y) and 2 chrominance parts (Cb and Cr) as shown in Figure 2a. The luminance Y, is commonly referred to as the grayscale image. The least significant bit of the center square grayscale image is then converted into a binary sequence via zig-zag or circular reading. The circular scan in this case is preferred to zigzag scan as shown in Figure 2b.

In Figure 3 a sample of webcam image is shown on the left. On the right, the least significant bit of the grayscale image appears to be noise bits. The binary sequence shall then be divided into several blocks of intended cryptographic keys and tested for their randomness using selected NIST random tests.

RANDOM NUMBER NIST TEST SUITE

The NIST Test Suite is a statistical package consisting of 16 tests that were developed to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence. Some tests are decomposable into a variety of subtests. The 16 tests are:

1. The Frequency (Monobit) Test,
2. Frequency Test within a Block,
3. The Runs Test,
4. Test for the Longest-Run-of-Ones in a Block,
5. The Binary Matrix Rank Test,
6. The Discrete Fourier Transform (Spectral) Test,
7. The Non-overlapping Template Matching Test,
8. The Overlapping Template Matching Test,
9. Maurer's "Universal Statistical" Test,
10. The Lempel-Ziv Compression Test,
11. The Linear Complexity Test,
12. The Serial Test,
13. The Approximate Entropy Test,
14. The Cumulative Sums Test,
15. The Random Excursions Test, and
16. The Random Excursions Variant Test.

Half of the tests are only suitable for binary sequence coming from pseudo-random number generator which consists of at least a million bits such as Mersenne Twister. Only 8 tests are particularly suitable for practical cryptographic keys size here. The selected NIST Test for short keys are listed in the Table 1 below.

TABLE 1: The list suitable random tests for short keys

0	Statistical Test	Min practical key size n
1	Frequency Monobit	128 bits
2	Block Frequency (M =128)	128 bits
3	Cumulative Sums (Forward and Backward)	128 bits
4	Runs	128 bits
5	Longest Runs of Ones	128 bits
6	Spectral DFT	1024 bits
7	Approximate Entropy (m = 4)	128 bits
8	Serial (m = 5)	128 bits

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[9]. The parameter α denotes the significance level that determines the region of acceptance and rejection. NIST recommends that α be in the range [0.001, 0.01].

A statistical test is formulated to test a specific null hypothesis (H_0). For the purpose of this document, the null hypothesis under test is that the sequence being tested is random against the alternative hypothesis (H_1) for which the sequence is not random.

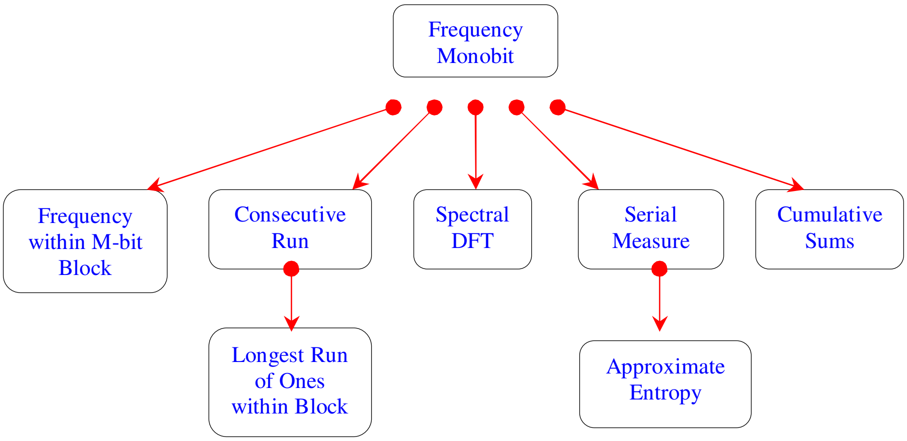

Figure 4a : The hierarchy of random tests : Failing higher test will ensure the failure of the lower test.

The 8 selected tests are basically relies heavily on the randomness of the binary sequence.

Figure 4a 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.

EXPERIMENTAL RESULTS

In this study, a sample has been captured using a typical webcam as shown in Figure 3. The common threshold value used to identify image background should be disabled. Preferably, the least significant bits are converted into one dimensional bits via circular reading instead of zigzag. The circular scan should start from the center of the image. The long binary sequence is then divided into blocks of 1024 bits. The first 10 block has been tested using the 8 NIST Statistical Tests suitable for short binary sequence. The results are shown in Table 3.

The result shown in the Table 3 below as the P-values of the tests is larger than required $\alpha = 0.01$ in order to reject the null hypothesis as random sequence. The P-values on 10 blocks of 1024-bit of the least significant bit of room noise coming from a user's webcam image in Figure 3 after going through circular reading passes all 8 NIST Statistical Tests.

TABLE 2: The Test Number of various NIST Statistical Tests on the least significant bit of room noise coming from a user's webcam image.

Test Number	Statistical Test
1	Frequency Monobit
2	Block Frequency (M = 8)
3a	Cumulative Sums (Forward)
3b	Cumulative Sums (Backward)
4	Runs
5	Longest Runs of Ones (M = 8)
6	Spectral DFT
7	Approximate Entropy (m = 7)
8a	Serial (m = 7) P-value1
8b	Serial (m = 7) P-value2

TALE 3: The results of 8 NIST Statistical Tests on 10 blocks of 1024-bit coming from a user's webcam image after going through circular reading.

Test #	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8	Block 9	Block 10
1	0.053	0.755	0.708	0.617	0.382	0.574	0.574	0.061	0.235	0.261
2	0.083	0.554	0.702	0.842	0.417	0.945	0.134	0.742	0.733	0.723
3a	0.053	0.422	0.601	0.494	0.494	0.160	0.804	0.113	0.171	0.130
3b	0.091	0.687	0.927	0.883	0.208	0.469	0.831	0.078	0.422	0.319
4	0.151	0.534	0.947	0.136	0.477	0.322	0.859	0.280	0.721	0.306
5	0.704	0.310	0.211	0.615	0.563	0.492	0.712	0.889	0.453	0.925
6	0.935	0.465	0.777	0.903	0.274	0.274	0.598	0.627	0.081	0.465
7	0.569	0.838	0.788	0.192	0.998	0.161	0.777	0.941	0.011	0.724
8a	0.430	0.222	0.360	0.461	0.764	0.050	0.757	0.233	0.028	0.797
8b	0.153	0.486	0.477	0.422	0.844	0.049	0.926	0.052	0.014	0.473
Average	**0.322**	**0.527**	**0.650**	**0.557**	**0.542**	**0.350**	**0.697**	**0.402**	**0.287**	**0.512**
Min	0.053	0.222	0.211	0.136	0.208	0.049	0.134	0.052	0.011	0.130
Max	0.935	0.838	0.947	0.903	0.998	0.945	0.926	0.941	0.733	0.925

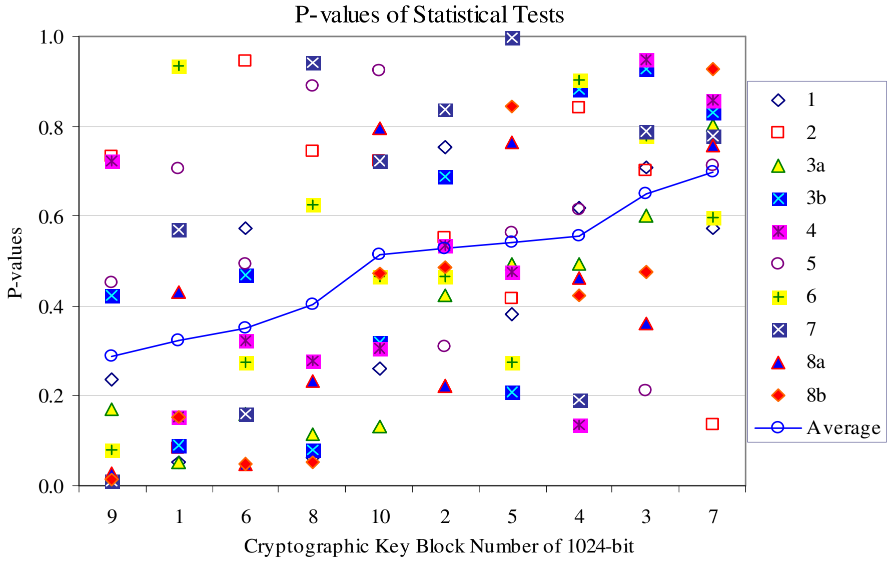

Figure 4b: The P-values are sorted according to the average scores out of the 10 statistical results for display

Based on the average score of the P-values, the results may sorted for better display. The sorted data is displayed in Figure 4b.

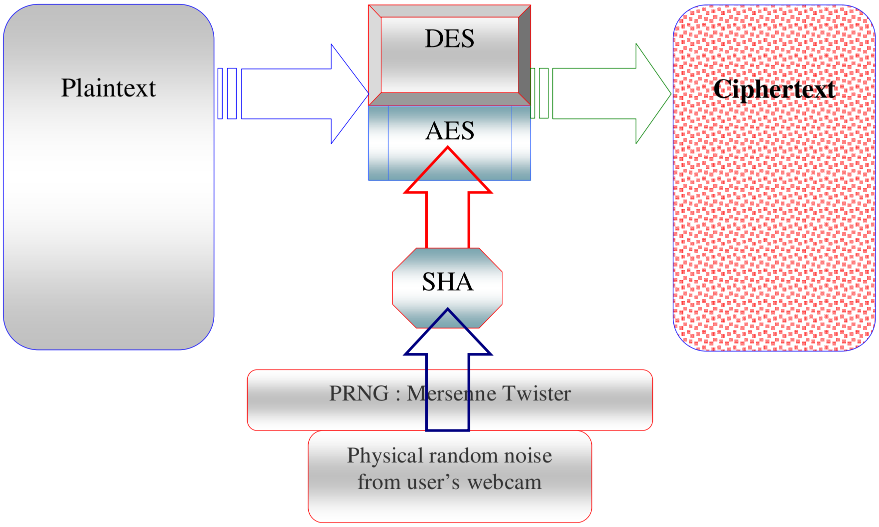

Figure 5: Physical random input should be taken as the primary source of session key or a seed of pseudo random number generator before going through hashing function.

For a basic cryptographic application in encrypting a plaintext file using symmetric encryption such as AES or DES, the random noise key should still be hashed. The key can also be used as seed of large pseudo random number generator such as Mersenne Twister as shown in Figure 5. The binary block with the highest average P-value scores shall be selected as random cryptographic key. In the example above, the 1024-bit block number 7 shall be selected for cryptographic use.

CONCLUSION

The aim of this paper is to show the usage of room ambience for generating true random bits. Due to difficulty of proving unpredictability in a theoretical way, the proposed true random bit is subjected to statistical tests. Due to the room ambience, the collections of least significant bits in the digital webcam have shown to pass all the NIST statistical tests.

Current digital webcam is capable of capturing high quality image. Thus, user's image from the webcam can be a good source of random cryptographic key. In this paper, the least significant bit of the ambience webcam image has been statistically proven to be random.

In practical cryptographic application it is imperative to produce short cryptographic key in a consistent manner. The block size of cryptographic key which is typically 128, 256, 1024 bits and so on.

This study has been designed to support practical cryptographic operations in the near future. A step-by-step procedure shall be developed to produce short 128, 256, 512 and 1024 bit keys. Selected NIST random tests shall be used. At least 100 candidate set of keys shall be generated per webcam image and test for randomness. The best candidate shall be produced based on the P-values of the selected NIST statistical random tests.

REFERENCES

[1] Tryggvi Björgvinsson, A Comparison of the Mersenne Twister and the Linear Congruential Method for Generating Pseudorandom Numbers, Chalmers University of Technology, 1 June 2005.

[2] Mads Haahr, True Random Number Service, http://www.random.org/

[3] John Walker, How HotBits Works, http://www.fourmilab.ch/hotbits/

[4] Landon Curt Noll, Simon Cooper, and Mel Pleasant, LavaRnd is a Random Number Generator, http://www.lavarnd.org/

[5] Fabian Monrose, Michael K. Reiter, Qi Li and Susanne Wetzel, Cryptographic Key Generation from Voice, *Proceedings IEEE Symposium on Security and Privacy, S&P 2001*, Oakland, CA , 14-16 May 2001, pp.202-213

[6] B. Chen and V. Chandran, Biometric Based Cryptographic Key Generation from Faces, *Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications DICTA 2007, IEEE Computer Society*, December 2007. pp394-401.

[7] Nur Azman Abu and Zulkiflee Muslim, Random Room Noise for Cryptographic Key, *Proceedings 2nd IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008*), Phitsanulok, THAILAND, 27-29 February 2008, pp381-387.

[8] Nur Azman Abu and Zulkiflee Muslim. Random Number Generation for Cryptographic Key, *Proceedings International Conference on Engineering and ICT(ICEI 2007)*, Melaka, MALAYSIA, 27-28 November 2007, Volume 1, pp 255-260.

[9] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo, A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications, NIST Special Publication 800-22, 15 May 2001.