Fingerprint Image for the Randomness Algorithm

Jong-Min Park, *Member, KIMICS*

***Abstract* – We present a random bit generator that uses fingerprint image for the source of random, and random bit generator using fingerprint image for the randomness has not been presented as yet.**

Fingerprint image is affected by the operational environments including sensing act, nonuniform contact and inconsistent contact, and these operational environments make FPI to be used for the source of random possible. Our generator produces, on the average, 9,334 bits a fingerprint image in 0.03 second. We have used the NIST SDB14 test suite consisting of sixteen statistical tests for testing the randomness of the bit sequence generated by our generator, and as the result, the bit sequence passes all sixteen statistical tests.

***Index Terms* – fingerprint, image, random bit generator, algorithm**

Ⅰ. INTRODUCTION

The need for random and pseudo random bit sequence arises in many cryptographic algorithms and also in many cryptographic protocols.

Two basic types of generator are used to produce random bit sequence: (1) random bit generators(RBGs); (2) pseudo random bit generators(PRBGs).

RBGs are divided into software methods and hardware methods. Software methods use the information in computer itself [1] or the information occurring while user and computer interact [2]. The shortcomings of software methods are that an adversary can manipulate the processes depending on the computer platform, and an user feels inconvenient. Hardware methods are based on physical phenomena that in themselves a portion of unpredictability, and suggested for avoiding shortcomings of software methods [3]-[5]. But the hardware methods bear their own shortcomings such as difficulty in implementation and the need of the extra device.

RBGs use linear congruential function (LCG), one way function (OWF) or trapdoor OWF. While LCGs are commonly used for simulation purposes and probabilistic algorithms, and pass the statistical tests, they are predictable and hence insecure for cryptographic purposes.

Manuscript received August 19, 2010; revised September 28, 2010; accepted October 1, 2010.

Authors are with the Cyber Security, Chosun University College of Science & Technology, Gwangju, 501-744, Korea(Email: pjm5234 @lycos.co.kr)

It has been proven that if OWF or trapdoor OWF exist, then, given a random seed, it is possible to generate more randomness than RBG [6],[7]. PRBGs using OWF have better performance in the aspect of execution time and generation rate, than RBGs [8],[9]. Trapdoor OWF is slow compared to OWF, therefore trapdoor OWF based PRBGs [10],[11] are used in some restricted circumstances. The limitation of PRBGs is that the random seed must be secure.

Fingerprint image (FPI) is affected by the operational environments including sensing act, nonuniform contact and inconsistent contact [12],[13]. The operational environments make a point of a finger to be sensed at different locations on the fingerprint acquisition device and to be represented by different pixel values within FPI. For these reason, we think that FPI can be used for the randomness.

In this paper, we present an algorithm that uses FPI for generating a random bit sequence. Our generator determines a set of pixel values and produces a bit from the pixel values in the set. To determine the set, we have made several experiments including the average frequencies of pixel values. The set determined in our generator contains at least one pixel value such that differs from the eight neighboring pixel-values with high probability.

The device of optical prism method is used to acquire gray level FPI of which size is 292×248. The generator has been implemented in C++ running on Windows XP server on a 800 MHz Pentium Ⅳwith 512 Mbytes of RAM. The generator produces, on the average, 9,334-bits a FPI in 0.03 second. The NIST test suite [14] consisting of sixteen traditional statistical tests is used for testing the randomness of the generated bit sequence, and, as the result, the generated bit sequence passes all statistical tests.

This paper is organized as follows. Section 2 includes notations and terminology. In Section 3, we examine FPI as the source of random, and in Section 4, we describe our generator. Section 5 includes the experimental result, and we conclude this paper in Section 6.

Ⅱ. PRELIMINARIES

The information carrying features in a fingerprint are the line structures called ridges and valleys. At FPI of Fig. 1 (a), the ridges are black and the valleys are white.

(a) FPI (b) Binarized FPI (c) runs

Fig. 1 FPI, binary FPI and runs

A gray level FPI can be considered as two dimensional array consisting of the pixel values represented by 8-bits. T[H][W] denotes the gray level FPI of which size is H(292)*W(248), and T[i][j] denotes a pixel value at jth column of ith row in T[H][W] for 0≤i≤H-1 and 0≤j≤W-1.

Binarization of FPI is to map a pixel value at the ridge into 1 and a pixel value at the valley into 0. Several thresholding methods including [15],[16] have been proposed to binarize FPI. $B[H][W]$ represents the binarized FPI of $T[H][W]$, and $B[i][j]$ denotes a pixel value at j th column of i th row in $B[H][W]$ for $0 \le i \le H-1$ and $0 \le j \le W-1$. Fig. 2(b) shows $B[H][W]$ of $T[H][W]$ at Fig. 2(a). To obtain $B[H][W]$, we have applied local thresholding method to $T[H][W]$ such that maps $B[i][j]$ into 1 if $B[i+k][j+l]/25 \le B[i][j]$ and $B[i][j]$ into 0 if otherwise for $1 \le k, l \le 5$.

Each column of $B[H][W]$ consists of 0's run(s) and 1's run(s). R_i denotes the number of runs at i th column of $B[H][W]$, N_j^i the length of j th run, S_j^i the starting row of j th run, and E_j^i the ending row of j th run. We divide 1's run into ridge run and trapezoid run, and 0's run into valley run and trapezoid run. Table 1 and Table 2 show the ranges of m where $B[m][i]$ is contained in ridge, trapezoid or valley run.

TABLE 1.
RANGE OF ROWS CONTAINED IN RIDGE, TRAPEZOID OR VALLEY RUN FOR $j=1$ OR $j=R_i$

N_j^i ($j=1$ or $j=R_i$)	Trapezoid run	Ridge or valley run
$N_j^i=1$	$S_j^i \le m \le S_j^i$	No m
$N_1^i \ne 1$	$S_j^i+2\times\lfloor N_j^i/4 \rfloor \le m \le E_j^i$ $S_j^i+2\times\lfloor N_j^i/4 \rfloor \le m \le E_j^i$	$S_j^i \le m \le S_j^i+2\times\lfloor N_j^i/4 \rfloor -1$
$N_R^i \ne 1$	$S_j^i \le m \le S_j^i+2\times\lfloor N_j^i/4 \rfloor$	$S_j^i+2\times\lfloor N_j^i/4 \rfloor +1 \le m \le E_j^i$

TABLE 2
RANGE OF ROWS CONTAINED IN RIDGE, TRAPEZOID OR VALLEY RUN FOR $2 \le j \le R_i-1$

N_j^i ($2 \le j \le R_i-1$)		Trapezoid run	Ridge or valley run
$4\times l$		$S_j^i \le m \le S_j^i+\lfloor N_j^i/4 \rfloor -1$ and $S_j^i+3\times\lfloor N_j^i/4 \rfloor \le m \le E_j^i$	$S_j^i+\lfloor N_j^i/4 \rfloor \le m \le S_j^i+3\times\lfloor N_j^i/4 \rfloor -1$
$4\times l+1$	$i \ne 0$	$S_j^i \le m \le S_j^i+\lfloor N_j^i/4 \rfloor -1$ and $S_j^i+3\times\lfloor N_j^i/4 \rfloor +1 \le m \le E_j^i$	$S_j^i+\lfloor N_j^i/4 \rfloor \le m \le S_j^i+3\times\lfloor N_j^i/4 \rfloor$
	$i=0$	$S_j^i \le m \le S_j^i$	No m
$4\times l+2$	$i \ne 0$	$S_j^i \le m \le S_j^i+\lfloor N_j^i/4 \rfloor -1$ and $S_j^i+3\times\lfloor N_j^i/4 \rfloor +1 \le m \le E_j^i$	$S_j^i+\lfloor N_j^i/4 \rfloor \le m \le S_j^i+3\times\lfloor N_j^i/4 \rfloor$
	$i=0$	$S_j^i \le m \le S_j^i$ and $E_j^i \le m \le E_j^i$	No m
$4\times l+3$	$i \ne 0$	$S_j^i \le m \le S_j^i+\lfloor N_j^i/4 \rfloor -1$ and $S_j^i+3\times\lfloor N_j^i/4 \rfloor +2 \le m \le E_j^i$	$S_j^i+\lfloor N_j^i/4 \rfloor \le m \le S_j^i+3\times\lfloor N_j^i/4 \rfloor +1$

Ⅲ. FINGERPRINT IMAGE AS THE SOURCE OF RANDOM

Various operational environments affect to FPI [7,10], and three operational environments that are related to our generator are listed below.

(1) Inconsistent contact: the act of sensing distorts the finger. Determined by the pressure and contact of the finger on the glass platen, the three dimensional shape of the finger gets mapped onto the two dimensional surface of the glass platen. Typically, this mapping function is uncontrolled and results in different inconsistently mapped FPIs across the impressions.

(2) Nonuniform contact: the ridge structure of a finger would be completely captured if ridges of the part of the finger being imaged are in complete optical contact with the glass platen. However, the dryness of the skin, skin disease, sweat, dirt, and humidity in the air confound the situation, resulting in a non-ideal contact situation: some parts of the ridges may not come in complete contact with the pattern, and regions representing some valleys may come in contact with the glass platen. This results in "noisy" low-contrast images.

(3) Sensing act: the act of sensing itself adds noise to the image. For example, residues are leftover from the previous fingerprint capture. A typical finger imaging system distorts the image of the object being sensed due to imperfect imaging conditions. In automated fingerprint identification system sensing scheme, for example, there is a geometric distortion because the image plane is not parallel to the glass platen.

Above operational environments make a point of a finger to be represented by different pixel values within FPIs of a finger and also to be sensed at different location

of the FPI acquisition device, and for these reasons, FPI can be used for the randomness. Fig. 2 show three FPIs of a finger.

Fig. 2 Three FPIs of a finger

Our generator does not produce a pixel value as a random number, because the frequencies of the pixel values are not equal reliably. Fig. 3 shows, on the average, the frequencies of the pixel values for our test 1,000 T[H][W]s of a finger.

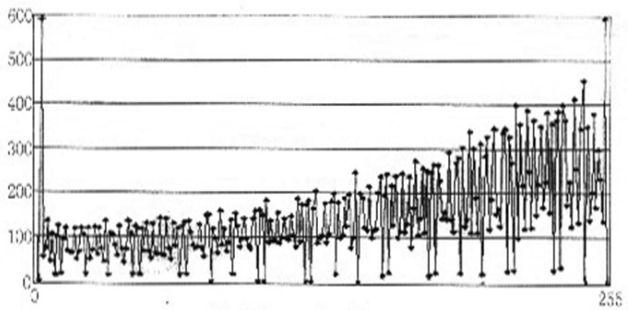

Fig. 3 Frequencies of pixel values

Our generator also does not produce a bit from a pixel value, because the probability of $T[i][j] = T[i \pm m][j \pm n]$ is greater than 1/8 reliably ($-1 \le m, n \le 1, 1 \le i \le H-2$ and $1 \le j \le W-2$). For our test 1,000 FPIs of a finger, the number of $T[i][j] \ne T[i \pm m][j \pm n]$ is, on the average, 5.017 for $-1 \le m, n \le 1$, $1 \le i \le 290$ and $1 \le j \le 246$.

From the results of two experiments mentioned above, we think that RGB using FPI has to product a bit from a set of the pixel values.

IV. RANDOM BIT GENERATOR USING FINGERPRINT IMAGE

Our generator determines a set consisting of the indices of [Rj/I] rows where 1<I< Rj , and denotes the ceiling function, and the set is determined by the following equation:

$$\begin{cases} \left\lfloor \frac{2 \times S_{Yk+1} - N_{Yk+1}}{2} \right\rfloor, & I = \frac{R-1 \bmod I}{2} \text{ and } 0 \le k \le \left\lfloor \frac{R-1}{I} \right\rfloor, \text{ if } (R_j - 1) \bmod I \ne 0, \\ \left\lfloor \frac{2 \times S_{Yk+1} - N_{Yk+1}}{2} \right\rfloor & \text{for } 1 \le k \le \frac{R-1}{I} - 1, Ek >, \text{ otherwise.} \end{cases}$$

Our generator produces a bit from the ser of pixel values denoted by {T[i][i][Yk≤i≤Yk-1]} where Yk is kth index in the set determined by above equation. The generator concatenates all of the pixel values in the set {T[i][i][Yk≤i≤Yk-1]} and then produces even parity bit of the concatenated bit string as a random bit. The generator, called RBG using FPI, is depicted in the follow.

Algorithm RBG using FPI
Input : T[H][W] and I.
Output : A random bit sequence generated from T[H][W].
begin
Step 1: Get B[H][W]
Step 2: for i=0 to W-1 do
Step 2.1 : Let { Yj, 1≤j≤n} be the set of the indices of rows.
Step 2.2 : for j=1 to n-1 do
Step 2.2.1 : Concatenate T[k][i] where j≤k≤Yj+1-1.
Step 2.2.2 : output even parity bit of the bit string generated in Step 2.2.1.
Step 2.3 : Concatenate T[k][i] where Yn-1≤k≤Yn.
Step 2.4 : output even parity bit of the bit string generated in Step 2.3
end

Fig.4 shows the figures obtained in the execution of RBG using FPI. At Fig.4 (a), I's run is black and O's white.

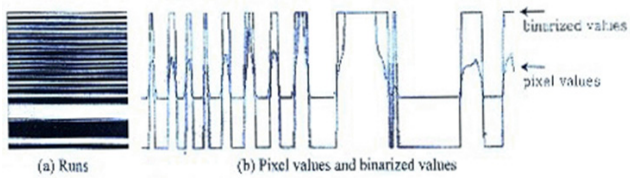

Fig. 4 Images obtained in the execution of RBG using FPI

In the algorithm RBG using FPI, each ser of pixel values producing a bit contains at least one pixel Value at the trapezoid run. For our test 1,000 FPIs of finger. the number of T[i][j] ≠T[i±m][j±n] is, one the average, 7.06 for -1≤m, n≤1, 1≤i≤290 and 1≤j≤246, when B[i][j] is located at trapezoid run.

V. EXPERIMENTAL RESULTS

A. Performance

Related to the performance, we consider the measures: (1) generation rate; (2) easy of implementation; (3) convenience of use; (4) execution time; (5) static memory requirement; (6) need of extra device.

Our generator had processed FPI, on the average, in 0.03 second for each I. and needs static memory of 1.34MB, Table 3 shows the averages of the generation rates for our test 1,000 T[H][W]s of a finger.

The generation rates are superior to RBGs using the information occurring while user and system interact. and generation rate per second is also superior to hardware based RBGs and software based RBGs using the information in computer itself.

TABLE 3.
AVERAGED OF THE GENERATION RATES

I	1	2	3	4	5
Average generation rates	9,324	4,701	3,218	2,277	1,986

Our generator was implemented easily, because the implementation of our generator is not system level but application level. The generator needs one touch to the device to obtain FPI. This means that our generator is more convenient than RBGs using the information occurring while user and system interact, but less convenient than hardware based RBGs and software based RGBs using the information in computer itself. Our generator has unavoidable shortcoming that is the use of the FPI acquisition device. But biometrics is a rapidly evolving technology that has been widely used in forensics, such as criminal identification and prison security, and in a very broad range of civilian applications, and the is being cheaper and cheaper, and in addition to, our generator is most adaptable to the automated fingerprint verification system needing RBG.

B. statistical tests

The NIST test suite is a package consisting of sixteen statistical tests that were developed to test the randomness of binary sequences produced by either RBG or PRBG [12]. These tests focus on a variety of different types of non-randomness that could exist in a sequence.

In the NIST test suite, the level of the recommended significance is 0.01% for all tests, and p-value is compared to 0.01, and the bit sequence passes a test, then the p-value is greater than 0.01. To obtain the size of the required bit sequence, we have concatenated the bit sequences produced from the FPIs or truncated a bit sequence produced from a FPI. Table 4 shows the results of the sixteen statistical tests for the bit sequence generated by RBG using FPI for a case I=1.

TABLE 4.
RESULTS OF THE STATISTICAL TESTS

Test	Length of bit sequence	p-value
Monobit test	1000	0.534146
Frequency test within a block	1000	0.739918
Runs test	1000	0.862137
Cumulative sums test	1000	0.739918(forward) 0.534146(reverse)
Discrete Fourier Transform test	1000	0.350485
binary matrix rank test	1000000	0.122325
The longest run of ones in a block	750000	0.122325
Non-overlapping template matching test	1000000	0.911413
Overlapping template matching test	1000000	0.213309
Maurer's Universal Statistical test	1000000	0.739918
Lempel-Ziv compression test	1000000	0.066382
Linear complexity test	1000000	0.075148
Serial test	1000000	0.739918
Approximate entropy test	1000	0.122325
Random excursions test	1000000	0.098912
Random excursions variant test	1000000	0.102833

VI. CONCLUSIONS

The device of optical prism method is used to acquire gray level FPI of which size is 292X248, RBG using FPI has been implemented in C++ running on Windows XP server on a 800 MHz Pentium Ⅳwith 512 Mbytes of RAM for estimating the performance, and we have used NIST SDB14 test suite to test the Randomness of binary sequences produced by RBG using FPI.

we have proposed fingerprint image as a new source of random and presented an algorithm called RBG using FPI generating a random bit sequence from a finger print image. Our generator is excellent in the aspect of generation rate, execution time, memory requirement and easy of implementation. The bit sequence generated by

RBG using FPI passes all sixteen statistical random tests in test suite offered by NIST SDB14.

REFERENCES

[1] RSA data Security, “Inc-RSA secure PC for windows 55 Users Manual.” 1997

[2] J. B. Lacy, D. P. Mitchell, and W. M. Schell, “Cryptolib: Cryptography in software,” In USENIX Security Symposium IV Proceedings, USENIX Association, pp.1-17, 1993.

[3] G. B. Agnew, ”Ransom sources for cryptographic systems, ”Eurocrypt ’87, Springer-Verlag, LNCS v.304, pp.77-81, 1988.

[4] D. Davis, R. Ihaka, and P. Fenstermacher, “Cryptographic randomness from air turbulence in disk drives,” Crypto ’94, Springer-Verlag, LNCS v.839, pp.114-120, 1994.

[5] M. Jakobsson, E. Shriver, B. K. Hillyer, and A. Juels, “A practical secure Random bit generator,” ACM Conference on Computer and Communications Security, pp.103-111, 1998.

[6] J. Hastad, “Pseudo random number generators under uniform assumptions,” In Proceeding of the Twenty Second Annual ACM Symposium on Theory of Computing, pp.395-404, 1990.

[7] M. Luby, Pseudorandomness and Cryptographic Applications, Princeton University Press, New Jersey, 1996.

[8] FIPS 186, “Digital signature standard,” Federal Information Processing Standards 186, U.S. Department of Commerce/NIST, National Technical Information Service, Springfield, 1994.

[9] A. Shacur, “Or the generation of cryptographically strong pseudorandom sequences” ACM Transactions on Computer Science, pp 38-44, 1983.

[10] L. Blum, M. Blum, and M. shub, “A simple unpredictable pseudo random generator,” SIAM Journal on Computing, 15, pp.364-383, 1986.

[11] S. Micali and C. P. Schnorr, “Efficient perfect polynomial random number generators,” Journal of Cryptology, 3, pp.157-172, 1991.

[12] L. Hong, Y. Wan and A. K. Jain, “Fingerprint enhancement: algorithm and performance evaluation,” IEEE Trans. Pattern Anal. Mach. Intell. 20, p777, 1998.

[13] D. Mario and D. Maltoni, “Direct Gray-Scale Minutie Detection In Fingerprints,” IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, no.1, pp.27-40, 1997.

[14] NIST special prosecution 800-12. A statistical test suite for random and pseudorandom number generators for cryptographic applications. 2000

[15] M. R. Verma., A.K. Maynrklay, and B coatterjcee, “Edge ceteoction in fingerprints.” Pattern Recogmitiom.20.p513, 1987

[16] D. M. Weher “A cost effective fingerprint verification algorithm for commercial application.” Proceedings of the south African Syraposuno on Commendation and Signal Processing, p.9, 1992.

Jong-Min Park He received the A.I and Ph.D. degrees in the Dept. of computer Engineering from Chosun University. In 2008 he joined the faculty of Chosun University College of Science & Technology. His research interests information security, bio metrics, Network Security, Information Security, pattern recognition, artificial intelligence.