디지털 지문 이미지를 잡음원으로 사용하는 안전하고 효율적인 난수 생성기

박승배[†] · 주낙근[††] · 강문설[†††]

요 약

본 논문에서는 디지털 지문 이미지를 잡음원으로 하는 난수 생성기를 제안한다. 생체 정보를 잡음원으로 하는 난수 생성기는 아직까지 세계적으로 제안되지 않고 있다. 제안한 난수 생성기는 한 지문에 대하여 평균 9,334 비트를 0.03초에 생성하며, 생성된 비트 열은 NIST에서 권장한 16개의 난수성 통계 검증들을 모두 통과하였다.

Practically Secure and Efficient Random Bit Generator Using Digital Fingerprint Image for The Source of Random

Seung-Bae Park[†] · Nak-Keun Joo[††] · Moon-Seol Kang[†††]

ABSTRACT

We present a random bit generator that uses fingerprint image as the source of random, and the random bit generator is the first generator in the world that uses biometric information for the source of random in the world. The generator produces, on the average, 9,334 bits a fingerprint image in 0.03 second, and the produced bit sequence passes all 16 statistical tests that are recommended by NIST for testing the randomness.

키워드 : 난수 생성기(Random Bit Generator), 잡음원(Source of Random), 지문 이미지(Fingerprint Image), 이진화(Binarization), 통계 검증(Statistical Random Test)

1. 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 generators 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 [13] or the information occurring while user and computer interact [8]. The shortcoming of software methods is that an adversary can manipulate the processes depending on the computer platform, or the users feel inconvenient. Hardware methods are based on physical phenomena that in themselves a portion of unpredictability, and suggested for avoiding shortcoming of software methods [1, 3, 6]. The shortcomings of hardware methods are that all of them are difficult to implement, and most of them need extra device.

PRBGs 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. 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 [5, 9]. PRBGs using OWF have better performance, in the aspect of execution time and generation rate, than RBGs. For these reasons, several approaches for OWF based PRBGs have been proposed [4, 14]. Trapdoor OWF is slow compared to OWF, therefore trapdoor OWF based PRBGs in [2, 11] are used in some restricted circumstances. The limitation of PRBGs is that the random seed must be secure.

This paper, we present a new RBG that uses fingerprint image (FPI) as the source of random, and the aim of our RBG is to combine the positive aspects of RBGs while avoiding their shortcomings without the need of extra device.

FPI is affected by the operational environments including sensing act, nonuniform contact and inconsistent contact [7,

[†] 종신회원 : 초당대학교 컴퓨터과학과 교수
[††] 종신회원 : 동신대학교 인터넷정보학과 교수
[†††] 종신회원 : 광주대학교 컴퓨터전자통신공학부 교수
논문접수 : 2002년 10월 23일, 심사완료 : 2003년 5월 6일

10]. 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.

Our generator determines a set of pixel values and then 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. In our RBG, the set contains at least one pixel value that differs from the values of the eight neighboring pixels with high probability.

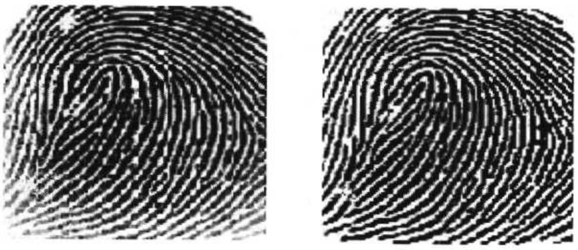
The device of optical prism method is used to acquire gray level FPI of which size is 292×248. The generator has been implemented in C++ running on Windows 2000 server on a 700MHz Pentium Ⅲ with 384Mbytes of RAM.

The generator produces, on the average, 9,334-bits a FPI in 0.03 second for the case that the number of pixel values in the set is smallest. The NIST test suite [12], 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. Section 4 describes our generator. Section 5 includes the experimental results. Section 6 concludes this paper.

2. Notations and Terminology

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



(a) FPI (b) binarized FPI

(Figure 1) FPI and binarized FPI of same finger

A gray level FPI can be considered as two dimensional array that contains pixel values represented by 8-bits. $T[H][W]$ denotes the gray level FPI of which size is $H \times W$, and $T[i][j]$ denotes a value of pixel at jth column of ith row at $T[H][W]$ for $0 \le i \le H-1$ and $0 \le j \le W-1$.

Binarization of FPI is to map pixel value at the ridge into 1 and pixel value at the valley into 0. Several thresholding methods have been proposed to binarize FPI [15, 16], and these methods have been used to discriminate ridge and valley in the automated fingerprint verification system. We applied to FPI local thresholding method to get the statistical values for FPI as the source of random. $B[H][W]$ represents the binarized FPI of $T[H][W]$, and $B[i][j]$ denotes a value of pixel at jth column of ith row at $B[H][W]$ for $0 \le i \le H-1$ and $0 \le j \le W-1$. (Figure 2) (b) shows $B[H][W]$ of $T[H][W]$ at (Figure 2) (a). To obtain $B[H][W]$, $B[i][j]$ is compared to $B[i+k][j+l]/25$ 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 ith column of $B[H][W]$, and N_j^i denotes the length of jth run at ith column. S_j^i denotes the starting row of jth run at ith column, and

〈Table 1〉 range of rows contained in ridge, trapezoid or valley run for $j=1$ or R_i

N_j^i ($j=1$ or R_i)	Trapezoid	ridge or valley run
$N_j^i = 1$	$S_j^i \le m \le S_j^i$	No m
$N_1^i \neq 1$	$S_j^i + 2 \times \lfloor N_j^i/4 \rfloor \le m \le E_j^i$	$S_j^i \le m \le S_j^i + 2 \times \lfloor N_j^i/4 \rfloor - 1$
$N_{R_i}^i \neq 1$	$S_j^i \le m \le S_j^i + 2 \times \lfloor N_j^i/4 \rfloor$	$S_j^i + 2 \times \lfloor N_j^i/4 \rfloor + 1 \le m \le E_j^i$

〈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	ridge or valley
$4 \times l$		$S_j^i \le m \le S_j^i + \lfloor N_j^i/4 \rfloor - 1$ and $S_j^i + 3 \times \lfloor N_j^i/4 \rfloor \le m \le E_j^i$	$S_j^i + \lfloor N_j^i/4 \rfloor \le m \le S_j^i + 3 \times \lfloor N_j^i/4 \rfloor - 1$
$4 \times l+1$	$l \neq 0$	$S_j^i \le m \le S_j^i + \lfloor N_j^i/4 \rfloor - 1$ and $S_j^i + 3 \times \lfloor N_j^i/4 \rfloor + 1 \le m \le E_j^i$	$S_j^i + \lfloor N_j^i/4 \rfloor \le m \le S_j^i + 3 \times \lfloor N_j^i/4 \rfloor$
	$l = 0$	$S_j^i \le m \le S_j^i$	No m
$4 \times l+2$	$l \neq 0$	$S_j^i \le m \le S_j^i + \lfloor N_j^i/4 \rfloor - 1$ and $S_j^i + 3 \times \lfloor N_j^i/4 \rfloor + 1 \le m \le E_j^i$	$S_j^i + \lfloor N_j^i/4 \rfloor \le m \le S_j^i + 3 \times \lfloor N_j^i/4 \rfloor$
	$l = 0$	$S_j^i \le m \le S_j^i$ and $E_j^i \le m \le E_j^i$	No m
$4 \times l+3$	$l \neq 0$	$S_j^i \le m \le S_j^i + \lfloor N_j^i/4 \rfloor$ and $S_j^i + 3 \times \lfloor N_j^i/4 \rfloor + 2 \le m \le E_j^i$	$S_j^i + \lfloor N_j^i/4 \rfloor + 1 \le m \le S_j^i + 3 \times \lfloor N_j^i/4 \rfloor + 1$
	$l = 0$	$S_j^i \le m \le S_j^i$ and $E_j^i \le m \le E_j^i$	$S_j^i + 1 \le m \le S_j^i + 1$

E_j^i denotes the ending row of *j*th run at *i*th column.

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.

3. Fingerprint Image as Random Source

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

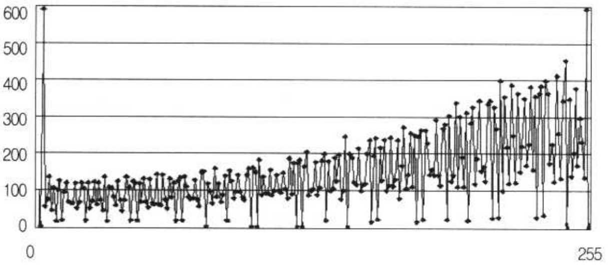
① Inconsistent contact : the act of sensing distorts the finger. Determined by the pressure and contact of the finger on the glass platen, the three dimensional shape of the finger gets mapped onto the two dimensional surface of the glass platen. Typically, this mapping function is uncontrolled and results 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 FPI and to be sensed at different locations of the FPI acquisition device. (Figure 2) shows three FPIs of same finger.

The operational environments make FPI to be used for the source of random possible, but our generator does not produce a pixel value as the random number because the frequencies of the pixel values are not equal for our test FPIs. (Figure 3) shows, on the average, the frequencies of the pixel values for 1,000 $T[H][W]$s of a finger.



(Figure 3) frequencies of pixel values

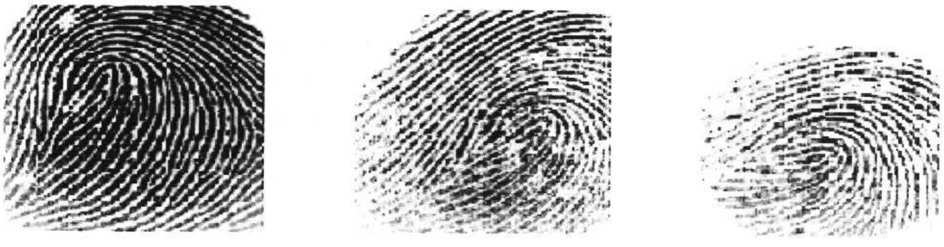
Our generator also does not produce a bit from a pixel value because the probability of $T[i][j] = T[i \pm m][j \pm n]$ is high for $-1 \leq m, n \leq 1$, $1 \leq i \leq H-2$ and $1 \leq j \leq W-2$. For our test 1,000 FPIs, the number of $T[i][j] \neq T[i \pm m][j \pm n]$, on the average, is 5.017 for $-1 \leq m, n \leq 1$, $1 \leq i \leq 290$ and $1 \leq j \leq 246$.

From the results of two experiments mentioned above, RBG using FPI for the source of random must produce a bit from a set of the pixel values.

4. Random Bit Generator using Fingerprint Image

Our generator uses Equation (4.1) to determine $\lceil R_j / I \rceil$ indices of rows for $1 \leq I < R_j$ where $\lceil \; \rceil$ denote the ceiling function. The generator uses a set of pixel values, denoted by $\{T[i][j] \mid Y_k \leq i \leq Y_{k+1} - 1\}$, to produce a bit where $\{Y_k \mid k$ is a positive integer$\}$ is the set of indices of rows.

Our generator concatenates all of the pixel values in the set $\{T[i][j] \mid Y_k \leq i \leq Y_{k+1} - 1\}$ and then produces even



(Figure 2) three FPIs of same finger

$$\begin{cases} \left\{ \left\lfloor \dfrac{2\times S^{i}_{I\times k+l+1}+N^{i}_{I\times k+l+1}}{2} \right\rfloor \,\middle|\, l=\left\lfloor \dfrac{R_i-1 \bmod I}{2} \right\rfloor \text{ and } 0\le k\le \left\lfloor \dfrac{R_i-1}{I} \right\rfloor \right\}, \text{ if } (R_i-1) \bmod I\ne 0. \\ \left\{0, \left\lfloor \dfrac{2\times S^{i}_{I\times k+1}+N^{i}_{I\times k+1}}{2} \right\rfloor \text{ for } 1\le k\le \dfrac{R_i-1}{I}-1, E_{R_i}\right\}, \text{ otherwise}. \end{cases} \quad (4.1)$$

parity bit of the concatenated bitstring 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 : 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 $\{Y_j \mid 1\le j\le n\}$ be the set of indices of rows
               determined by Equation (4.1).
    Step 2.2 : for $j$ = 1 to $n-1$ do
      Step 2.2.1 : concatenate $T[k][i]$ where $Y_j\le k\le Y_{j+1}-1$.
      Step 2.2.2 : output even parity bit of the bitstring generated
                   by Step 2.2.1.
    Step 2.3 : concatenate $T[k][i]$ where $Y_{n-1}\le k\le Y_n$.
  Step 2.4 : output even parity bit of the bitstring generated by
             Step 2.3.
end
```

(Figure 4) (a) shows the runs in a column where the runs of 1 are black and the runs of 0 are white, and (Figure 4) (b) shows 129 pixel values and the binarized values. Smoothing stage controls the number of pixel values in the set. That is, a set containing small number of pixel values is united to another set in smoothing stage.

In the algorithm RBG_using_FPI, the set of pixel values to produce a bit contains at least one pixel value $T[i][j]$ where $B[i][j]$ is at the trapezoid run. For our test 1,000 FPIs, the number of $T[i][j]\ne[i\pm m][j\pm n]$, on the average, is 7.06 for $-1\le m, n\le 1, 1\le i\le 290$ and $1\le j\le 246$ such that $B[i][j]$ contained in trapezoid run.

5. Experimental Results

To estimate the performance of our generator, 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, are considered.

The NIST test suite is a package consisting of 16 tests that were developed to test the randomness of binary sequences produced by either RBG or PRBG. These tests focus on a variety of different types of non-randomness that could exist in a sequence [12].

5.1 Performance

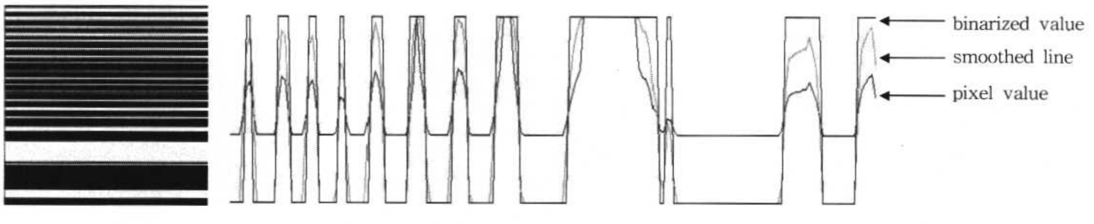
Our generator had processed FPI, on the average, in 0.03 second for each I, and needs static memory of 1.34MB. Therefore, our generator is also excellent for the execution time and the static memory requirement.

<Table 3> shows the averages of the generation rates for 1,000 $T[H][W]$s. 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〉 The averages of the generation rates

I	1	2	3	4	5
average generation rate	9,334	4,701	3,218	2,377	1,986

Our generator was implemented easily because the implementation of our generator is not system level but application level. Our 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, and less convenient than hardware based RBGs and software based RBGs using the information in computer itself.



(a) runs

(b) pixel values and binarized values of the pixel values

(Figure 4) images generated in the process of RBG_using_FPI

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. Therefore, the use of device for acquiring FPI is not serious shortcoming, and furthermore our generator is most suitable in the automated fingerprint verification system needing RBG.

5.2 Statistical Tests

<Table 4> shows the results for 16 statistical tests for a case $I = 1$. The level of significance is 0.01% for all tests, and p-value is compared to 0.01.

〈Table 4〉 Results of 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
Maurers 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

6. Conclusions

We have proposed fingerprint image as a new source of random, and presented an algorithm called RBG_using_FPI to produce a random bit sequence from fingerprint 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 16 statistical random tests recommended by NIST.

References

[1] G. B. Agnew, "Ransom sources for cryptographic systems," Eurocrypt '87, Springer-Verlag, LNCS Vol.304, pp.77-81, 1988.

[2] L. Blum, M. Blum and M. Shub, "A simple unpredictable pseudo random generator," SIAM Journal on Computing, 15, pp.364-383, 1986.

[3] D. Davis, R. Ihaka and P. Fenstermacher, "Cryptographic randomness from air turbulence in disk drives," Crypto '94, Springer-Verlag, LNCS Vol.839, pp.114-120, 1994.

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

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

[6] 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.

[7] L. Hong, Y. Wan and A. K. Jain, "Fingerprint enhancement : algorithm and performance evaluation," IEEE Trans. Pattern Anal. Mach. Intell. 20, p.777, 1998.

[8] 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.

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

[10] D. Mario and D. Maltoni, "Direct Gray-Scale Minutiae Detection In Fingerprints," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.19, No.1, pp.27-40, 1997.

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

[12] NIST Special Publication 800-12, A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2000.

[13] RSA Data Security, Inc., "RSA Secure PC for Windows 95 Users Manual," 1997.

[14] A. Shamir, "On the generation of cryptographically strong pseudorandom sequences," ACM Transactions on Computer Science, pp.38-44, 1983.

[15] M. R. Verma, A. K. Majumdar and B. Chatterjeee, "Edge detection in fingerprints," Pattern Recognition 20, p.513, 1987.

[16] D. M. Weber, "A cost effective fingerprint verification algorithm for commercial application," Proceedings of the South African Symposium on Communication and Signal Processing, p.9, 1992.

박 승 배

e-mail : sbpark@chodang.ac.kr

1989년 전남대학교 계산통계학과(이학사)

1992년 전남대학교 대학원 전산통계학과 (이학석사)

1996년 전남대학교 대학원 전산통계학과 (이학박사)

1996년~현재 초당대학교 컴퓨터과학과 조교수

관심분야 : 암호 알고리즘, 암호 프로토콜, 보안

주 낙 근

e-mail : nkjoo@mail.dsu.ac.kr

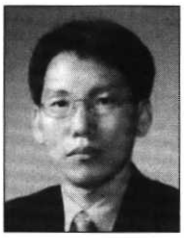
1985년 전남대학교 전산통계학과(이학사)

1987년 전남대학교 대학원 전산통계학과 (이학석사)

1995년 전남대학교 대학원 전산통계학과 (이학박사)

1991년~현재 동신대학교 인터넷정보학과 교수

관심분야 : 컴퓨터이론, 정보보안, 워터마킹



강 문 설

e-mail : mskang@hosim.kwangju.ac.kr

1986년 전남대학교 전산통계학과(이학사)

1989년 전남대학교 대학원 전산통계학과 (이학석사)

1994년 전남대학교 대학원 전산통계학과 (이학박사)

1989년~1994년 전남대학교 전산학과 조교 및 시간강사

1997년~2002년 한국정보처리학회 논문지 편집위원회(부위원장)

1994년~현재 광주대학교 공과대학 컴퓨터전자통신공학부 부교수

1996년~현재 한국정보처리학회 소프트웨어공학연구회 운영 위원회(편집위원)

관심분야 : 소프트웨어공학, 컴포넌트기반 소프트웨어 개발, 정보보호관리