Proceeding of the IEEE
International Conference on Automation and Logistics
Zhengzhou, China, August 2012

True Random Bit Generator Based on Cell Phone Recording and Chaotic Encryption

[1]Fei Xiang [2]Lei Zhang [2]Zhiyong Zhang and [3]Lili Zhang

[1]Department of Electronic Science & Technology, [2]Department of Computer,
[3]Department of Communication Engineering,
College of Electronic & Information Engineering,
Henan University of Science & Technology,
Luoyang, Henan 471003, P. R. C.

fayexiang@hotmail.com

Abstract – **This paper proposes a design method of true random bit generator(TRBG) based on cell phone recording and chaotic encryption. Firstly a recording from cell phone is collected, then is transformed the format operated by MATLAB. The recording is turned into a sequence in computer by MATLAB, which is encrypted by chaotic sequence from a chaotic system. Finally, through binarization a true random bit sequence is produced. Randomness and security analysis is carried out. Simulation results show that the properties of the bit sequence are excellent in randomness and security, and the method is safe, fast and easy to realize.**

Index Terms – Cell phone recording, true random, chaotic encryption.

I. Introduction

Random number has a very important role in the field of cryptography. The key of symmetric encryption algorithm must be generated randomly. When public encryption algorithm produces key pair, a large amount of random numbers is required. In addition, many digital signature algorithms and cryptographic protocols are required using random numbers.

In mathematics there are some usual methods to generate random numbers, such as RAND table, linear congruential generator and feedback shift register and so on. These sequences are decided by algorithms and seeds, where every value is determined if algorithms and seeds are decided. So these random numbers are called pseudorandom numbers. In physics the method is turning analog random signals in nature into discrete binary random bits by nonlinear transform systems. There are some usual random sources, such as natural noise [1] and quantum random events [2]. Yu [3] proposed to design true random number generator using piecewise linear chaotic system, which needed to build a analog circuit and the process was complex. Zhou [4] put forward to generate true random number using mouse movement and chaotic system, which used mouse movement as random source and utilized image encryption algorithm and Hash function to post-process mouse movement. The method was safe, fast, convenient and cheap. But with the development of PC technology, more and more electronic equipment adopt touch screen. The method depended on the mouse, which was not universal.

This paper brings forward using cell phone as random source, the reasons are as follows. 1) Cell phone is the most wide portable device. Almost all cell phones have the record function, which means getting sound signals does not need additional equipment. So the cost is reduced. 2) The collection of sound signals is not subject to the restriction of time and space, which is more convenient and more universal. 3) The security of this true random bit generator is higher, because the generation of random number is controlled by the user. However, as the entropy source the sound signals recorded anytime and anywhere have a very important defect that the sound signals have many similarities and patterns. If the similarities and patterns are not deleted in the post-process, the statistics of random numbers will be poor. On the other hand, even though two sounds are recorded by the same cell phone at the same place, the sounds can not be entirely same. Using this feature, if the output of post-processing is sensitive enough to the initial sound signal, the similarities will be deleted. It is well known that the chaotic system is sensitive to initial values and parameters and chaotic sequences have the properties of pseudo randomness and ergodicity. Therefore, the paper proposes using encryption of chaotic stream cipher to post-process the sound signals collected.

The rest of the paper is organized as follows. Section II introduces the design of TRBG based on cell phone recording and chaotic encryption. Section III gives the simulations and properties test. Section IV shows the application of the true random bit. At the end, section V concludes the algorithm and the results of simulation.

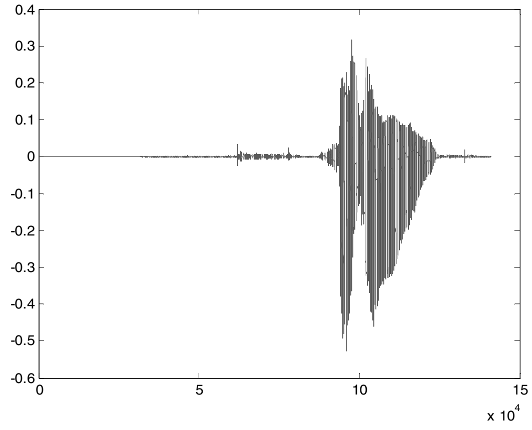
II. Design of TRBG

The design of TRBG based on recording of cell phone and chaotic encryption includes three steps. 1) Format transformation. Turn sound signal recorded by cell phone into the signal which is can be processed by PC. 2) Post-processing. Use encryption of chaotic stream cipher to remove the similarities and patterns in the sound signal. 3) Binarization. Transform real-value sequence into binary sequence. Next, the three steps will be described in detail.

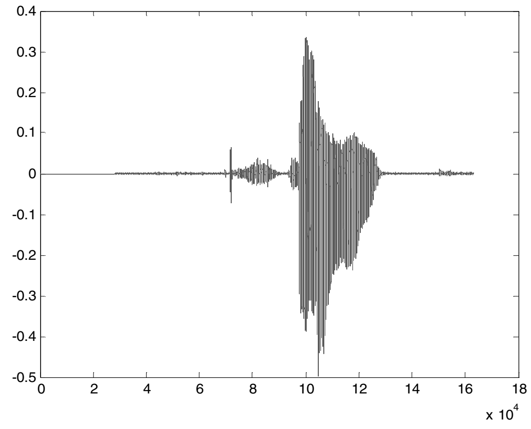
A. Format Transformation.

The voice recorded by cell phone is the '.amr' format, which is the abbreviation of adaptive multi-rate, proposed by European Telecommunications Standards Committee. It is the most widely used voice standards in the mobile communication system, and recognized as a format to save the recording of cell phone. The capacity of '.amr' file is very so small that the capacity of an audio file up to one minute is less than 50KB, which is a significant advantage for generation of large amount of random numbers. The format is '.wav', which is can be processed by MATLAB. So the file of '.amr' should firstly be turned into file of '.wav' in computer. We use the software of FormatFactory 2.45[5], which is software for format transformation of all kinds of multimedia files.

Fig. 1 shows two sound signals from the same person on the same place at the different time using the same cell phone. The person says "Hello", which lasts about 3 seconds. The .amr file is about 5K size, which is transformed to .wav file by FormatFactory 2.45. In Fig. 1 the horizontal axis expresses time, and the vertical axis expresses sound pressure after normalization. In MATLAB, the audio signal is read by the command of [y, Fs, bits]= wavread ("*.wav"), where y is the normalized sound pressure, 'Fs' is sampling frequency and bits is sampling bits. The default 'Fs' is equal 44100Hz, and the default 'bits' is equal 16. Read the audio signal of Fig. 1(a), a 141076-long sequence is obtained. Read the audio signal of Fig. 1(b), a 163126-long sequence is obtained.



(a)



(b)

Fig. 1 Sound signals

B. Post-processing.

As mentioned above, the sound by the same person may be similar in many aspects, which can be seen from Fig.1. In order to remove the similarity, the output must be sensitive to the sound pressure. Therefore the property can be used that chaotic system is extremely sensitive to the initial state and system parameters. The audio signal is encrypted by chaotic sequence. The range of the two sound signals are both [-1, 1] after they are read and processed by MATLAB. The length is 141076 and 163126, respectively. So the discrete chaotic maps can be used to generate the sequences of same length. The chaotic pseudorandom sequence can also be generated by continuous chaotic systems, using Runge-Kutta to solve the equations. Encryption can be realized using add or bit XOR after turning the sequence value into integer.

C. Binarization.

Using space partition method, the encrypted sequence can be transformed into binary sequence. Assume the encrypted sequence is $z(n)$. Let

$$s(n)=\begin{cases}0 & z(n)\in\bigcup_{d=0}^{2^{n-1}-1} I_{2d}^{n}\\ 1 & z(n)\in\bigcup_{d=0}^{2^{n-1}-1} I_{2d+1}^{n}\end{cases}$$

Where n is any positive integer, $I_0^n, I_1^n, \ldots, I_{2^n-1}^n$ represents 2^n continuous equal intervals in the range of $[a, b]$. $s(n)$ is the output binary sequence.

III. Simulations

A. Experiments Results.

The chaotic pseudorandom sequence is generated by the Logistic map

$$x(n+1)=4x(n)(1-x(n)).$$

Choose the initial value $x(1)$=0.001, the number of iterations is 165000, which is a little greater than the length of audio signal. Fig. 2 shows the figure of sequence, which displays the former 500 iterative values.

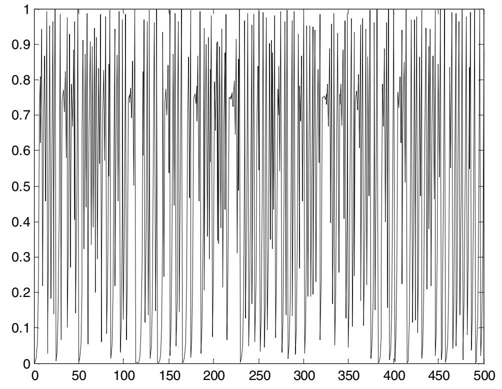
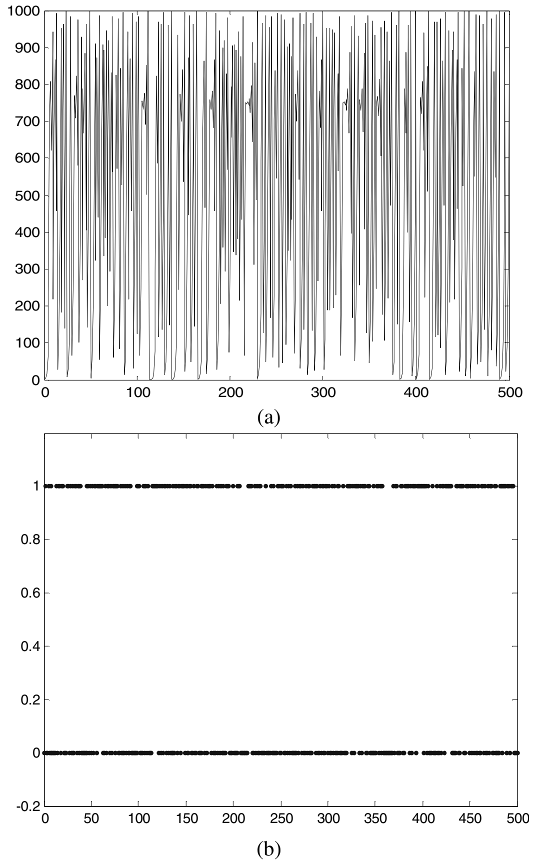


Fig. 2 Chaotic sequence the former 500 iterative value

Encrypt the audio signal sequence using two methods:

Method 1: $z(n)=K*x(n)+y(n)$, where $x(n)$ is the chaotic sequence, $y(n)$ is the audio sequence and K is the control parameter. In this method, the first recording is the random source, K=1000, which must be large enough to mask the audio sequence. The encrypted sequence $z(n)$ is shown as Fig. 3(a), and the binarization sequence is shown as Fig. 3(b). In order to see the two sequences clearly, we choose the former 500 values to show.

Method 2: $z(n)$=bitxor(floor(P*$x(n)$), floor(P*abs($y(n)$)))/P, where floor is the floor function, abs is the absolute value function, and P is the control parameter. Find the absolute


(a)

(b)

Fig. 3 Method 1:(a) Encrypted sequence $z(n)$; (b) Binarization sequence $s(n)$

value of $y(n)$ because the range of $y(n)$ is [-1,1] and We choose $P=2^{16}$ because the sampling bits is 16. The encrypted sequence $z(n)$ is shown as Fig. 4(a), and the binarization sequence is shown as Fig. 4(b).

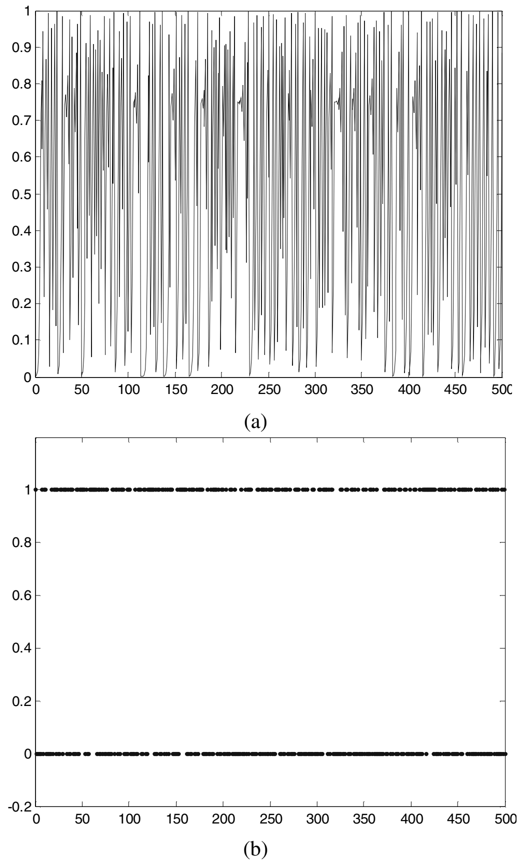
B. Speed.

TABLE I lists the average time required to generate the random bits using the two chaos-based methods.

TABLE I
THE AVERAGE TIME WITH THE TWO METHODS FOR SOUND SIGNAL A,B

Method	Method1		Method2	
	A	B	A	B
Total time(ms)	47	49	25	24
Number of random bits	16384	16384	16384	16384
Time to generate a bit(ms)	0.0029	0.0030	0.0015	0.0015
Speed(Kb/s)	340.43	326.53	640	666.67

The two methods are implemented with non-optimized MATLAB codes, running on an ordinary PC with a 2.00GHz Intel Core 2 Duo CPU. And the total time includes the time MATLAB reads the sound signal $y(n)$, the generation of $x(n)$, the calculation of $z(n)$ and the generation of $s(n)$ after binarization. It is observed that the speed is very fast. In literature [6], the time to generate a bit using hash function is 1.10ms. In literature [7], in all three chaos-based approaches, the shortest time to generate a 16384-bit random sequence is 1840ms. In literature [8], the 'MASK' approach cost 535ms to generate 256 bits. TABLE II compares the cost time of these approaches.


(a)

(b)

Fig. 4 Method 2:(a) Encrypted sequence $z(n)$; (b) Binarization sequence $s(n)$

TABLE II
COMPARISON WITH METHODS IN LITERATURES

Method	Method2 proposed	Ten map in Literature [6]	Arnold cat map Literature[7]	Mask in Literature [8]
Total time(ms)	49	92	1840	535
Number of random bits	16384	104	16384	256
Time to generate a bit(ms)	0.0030	0.88	0.1123	2.0898

C. Randomness Properties.

The US NIST statistical test suite[9] is used to test the randomness of the generated bits. It includes 16 statistical tests and each of them is formulated to test a null hypothesis that the sequence being tested is random. There is also an alternative hypothesis which states that the sequence is not random. For each test, there is an associated reference distribution, based on which a *P_value* is computed from the binary sequence. If this value is greater than a pre-defined threshold α(0.01 in default), the sequence passes the test. The two approaches that NIST has adopted include the examination of (1) proportion of sequences that pass a statistical test, and (2) uniformity of the distribution of those *P_value*.

1M random bits are generated by two methods using two sound signals shown as Fig.1. Those sequences are tested using US NIST test software. TABLE III lists the results of test results. In TABLE III, the 16 test names of US NIST statistical test are as follows: frequency test(FT), test for frequency within a block(FBT), runs test(RT), test for the longest run of ones in a block (LROBT), random binary matrix rank test(RBMRT), discrete Fourier transform test(DFTT), aperiodic template matching test(ATMT), periodic template matching test(PTMT), Maurer's universal test(MUST), Lempel-Ziv complexity text(LZCT), linear complexity test(LCT), serial test(ST), approximate entropy test(AET), cumulative sum test(CST), random excursions test(RET) and random excursions variant test(REVT).

TABLE III
THE TEST RESULTS

Test name	Method1		Method2	
	A	B	A	B
FT	0.6538	0.3892	0.2987	0.5749
FBT	0.2864	0.6839	0.3497	0.3983
RT	0.6576	0.6789	0.5465	0.5638
LROBT	0.3767	0.4586	0.4579	0.6754
RBMRT	0.6789	0.6694	0.3085	0.4590
DFTT	0.5656	0.4590	0.6794	0.4380
ATMT	0.4668	0.4589	0.4580	0.6734
PTMT	0.5465	0.3409	0.4770	0.3957
MUST	0.4546	0.4573	0.3396	0.6769
LZCT	1.0000	1.0000	1.0000	1.0000
LCT	0.2679	0.9734	0.4578	0.3730
ST	0.9786	0.5648	0.5487	0.9376
AET	0.6568	0.4648	0.5623	0.5503
CST	0.3646	0.4639	0.5759	0.3629
RET	0.8657	0.4299	0.4623	0.5397
REVT	0.8567	0.4638	0.5748	0.8392

It is shown as TABLE III that the random bits have passed all the tests of US NIST. So the randomness of those sequences is good enough to be used in the cryptograph.

IV. CONCLUSIONS

Random number generation is very important for cryptograph. This paper proposes a cheap convenient and universal method for generating random bits from cell phone recording. To overcome the similarity and patterns of recordings, we utilize chaotic encryption to post-process these sound signals. At last, we compare the cost time of our approach with that of other three approaches in literature [6, 7, 8], and test the randomness of those sequences using 16 statistical tests recommended by US NIST. Through the comparison and tests, the time required to generate the random bits is faster than that other two approaches, and those sequences have passed all 16 tests, which means that the methods are simple, fast and have excellent cryptographical properties.

ACKNOWLEDGMENT

The paper was supported by the National Natural Science Foundation of China (Grant No.61003234), Program for Science & Technology Innovation Talents in Universities of Henan Province (Grant No.2011HASTIT015), Technology Project Plan of State Archives Administration (Grant No. 2011-X-10) and Technology Project Plan of Henan Archives Administration (Grant No. 2011-X-37).

REFERENCES

[1] Sun J. W., Zhang S. L., Wang S. W.. Generation of high-speed true random bit sequence using white noise source. Acta Electronica Sinica, 2003, 31(8): 1255~1256
[2] Feng M. M., Qing X. L., Zhou C. Y., et al. Quantum random number generator based on polarization. A cta Physica Sinica, 2003, 52(1): 72~76
[3] Yu J., Shen H. B., Yan X. L.. Implementation of chaos-based high-speed truly random number generator. Chinese Journal of Semiconductors, 2004, 25(8):1013-1019
[4] Zhou Q., Liao X. F., Wong K. W., et al. True random number generator based on mouse movement and chaotic hash function. Information Sciences, 2009, 179: 3442-3450
[5] http://hongjin2.com/
[6] Zhou Q., Liao X. F., Wong K. W., et al. True random number generator based on mouse movement and chaotic hash function. Information Sciences, 2009, 179: 3442-3450.
[7] Zhao L., Liao X. F., Xiao D., et al. True random number generation from mobile telephone photo based on chaotic cryptography. Chaots, Solitons and Fractals, 2009, 42:1692-1699.
[8] Hu Y. Liao X. F., Wong K. K., et al. A true random number generator based on mouse movement and chaotic cryptography. Chaos, Solitons and Fractals, 2009, 40:2286-2293.
[9] NIST. A statistical test suite for random and pseudo-random number generators for cryptographic applications. http://csrc.nist.gov/rng/rng2.html, 2001.