EXPERIMENTAL VALIDATION OF PSEUDO TRUE RANDOM NUMBER GENERATION AND SYNCHRONIZATION USING NESTED LINEAR CHAOTIC MAPS BASED ON TMS320C6416

Q. Nasir, A. M. Abid and A. S. Elwakil
Electrical & Computer Engineering Department, University of Sharjah, Sharjah, U.A.E.

Keywords: Pseudo-true random bit generators, Chaotic maps, Chaos synchronization, NIST statistical test suite.

Abstract: A Pseudo True Random Binary Generator (PTRBG) based on a Nested Linear Chaotic Maps (NLCM) is proposed. Implementing the synchronization of chaotic systems presents a challenge. The paper proposes an implementation of a generation and synchronization method of PTRBG using NLCM and backward iteration synchronization approach. A prototype has been developed through Texas Instruments TMS320C6416 DSP development kit. Randomness tests of the generated bits of the PTRB is performed using the NIST statistical test suite.

1 INTRODUCTION

Random and pseudo-random numbers are used in many areas including test data generation, Monte-Carlo simulation techniques, generation of spreading sequences for spread spectrum communications, and cryptography(Ahmed and Siyal, 2006). Pseudo-random spreading sequences used in spread spectrum communications must be repeatable, while for most simulations using random numbers repeatability is not necessary. In cryptographic and security applications depends on the randomness of the source and the unpredictability of the used random bits(Tang and Tang, 2005). Various encryption techniques for secure transmission have been studied. The approaches include time domain scrambling techniques (Tang and Tang, 2005), and permutation and depermutation of Fast Fourier Transform (FFT) coefficients (Ahmed and Siyal, 2005). In recent years many researchers have noticed a close relationship between chaos and cryptography. Chaos appeared to be another paradigm to protect data and seems to be promising in the areas of security and cryptography. Chaos based encryption techniques such as in (Tang and Tang, 2005), (Drutarovsk and Galajda, 2006) are considered practical because they provide a good combination of speed, high security, complexity, reasonable computational overheads.

Chaotic circuits represent an efficient alternative to classical TRBG (Drutarovsk and Galajda, 2006). Studies in nonlinear dynamics show that many of the seemingly complex systems in nature are described by relatively mathematical equations(Sprott, 2003). Although chaotic systems appear to be highly irregular, they are also deterministic in the sense that it is possible to reproduce them with certainty. These promising features of chaotic systems attracted many researchers to try chaos as a possible medium for secure communication.The nonlinear phenomenon of chaos poses a promising alternative for pseudo-random number generation due to its unpredictable behaviour.

The chaotic system generates unpredictable pseudo random orbits which can be used to generate TRNGs (True Random Number Generators). Many different chaotic systems have been used to generate TRNGs such as Logistic map (Sajeeth et al., 2001), and its generalized version (Matthews, 1989), Chebyshev map, (Ahmed and Siyal, 2006) piecewise linear chaotic maps (NLCM) (Masuda and Aihara, 1999) and piecewise nonlinear chaotic maps (Tao et al., 1999). Chaotic systems are characterized by a sensitivity dependence on initial conditions, and with such initial uncertainties, the system behaviour leads to large uncertainty after some time.

A TRBG produce long sequences made of perfectly independent bits and when restarted, it never reproduces a previously delivered sequence. To assess the statistical properties and investigate the randomness of the TRBGs, several test suites are available such as AIS 21, AIS 31, FIPS 140 and the NIST statistical suite (NIST, 2001). TRBGs are usu-

Nasir Q., M. Abid A. and S. Elwakil A..
EXPERIMENTAL VALIDATION OF PSEUDO TRUE RANDOM NUMBER GENERATION AND SYNCHRONIZATION USING NESTED LINEAR CHAOTIC MAPS BASED ON TMS320C6416.
DOI: 10.5220/0003310001450150
In *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2011), pages 145-150
ISBN: 978-989-8425-48-5

ally difficult to implement; chaos-based random number generators implemented in the software provide a good method to produce pseudo-true random numbers. Chaotic systems or discrete chaotic maps are random like and deterministic but unpredictable in the long term. Consequently, the evaluation of chaotic systems in the field of cryptography has been extensively studied for several decades. This is due to the highly unpredictable and random-look nature of chaotic signals. Chaotic signals are aperiodic and exhibit sensitive dependence on initial values. Since they are governed by one or more control parameters, a small perturbation in these parameters can cause a large change in the state of the system. These features make chaos certainly suitable for applications that need a high level of security.

Chaotic streams can be generated using a number of chaotic maps such as Logistic map, 2-D Henon map, Chebyshev map, piecewise linear chaotic maps (PWLCM), piecewise nonlinear chaotic map, etc (Li, 2003). With two chaotic systems communicating over a noisy channel, the problem of synchronization becomes an essential part to study. In a communication system that uses chaos for encryption; synchronization is the ability of the receiver to recover the original message transmitted. This can be achieved only if the transmitter and receiver have the same copy of the chaotic binary sequence. For this reason several synchronization techniques exist in literature (Millerioux and Mira, 2001)- (Lau, 2006).

In spite of the extreme sensitivity to initial conditions of chaotic systems, synchronization can still be achieved. In (Millerioux and Mira, 2001), the observer synchronization method implemented depends on continuously feeding the chaotic system at the receiver with the error (difference) between the original chaotic sequence at the transmitter and the estimated sequence at the receiver . The impulsive synchronization technique in (Yong-Ai and Yi-Bei, 2002) involves the use of small control impulses where the chaotic maps must be asymptotically stable. According to (De Angeli et al., 1995), using the dead beat synchronization method, the synchronization error reach zero in exactly two steps. Another synchronization method was proposed in (Min et al., 2006), where synchronization is achieved by mixing discrete chaotic signals and using the output to drive the chaotic systems at the transmitter and the receiver. Synchronization using backward iterations and analysis of chaotic systems using symbolic dynamics was introduced in (Cong et al., 1999). In this method a number of backward iterations from a random initial condition are sufficient to reproduce an exact copy of the chaotic signal produced at the transmitter. This type of PTBGs can be easily included as part of Software Defined Radios (SDRs).

In this paper an implementation of software generation and synchronization of the chaotic binary sequences generated by NLCM is achieved by using the backward iteration synchronization approach (Cong et al., 1999), (Stojanovski and Kocarev, 1997). An implementation test bed for a complete communication system based on a TMS320C6416 DSP development kit is then described. Randomness of the pseudo true random bits (PTRB) generated is assessed using the NIST statistical test suite.

The rest of the paper is organized as follows. An overview of the main blocks in the chaotic communication system proposed is given in Section II. Section III describes the TRBGs studied. The chaotic synchronization method is explained in Section IV. The results of subjecting the bit sequences to NIST randomness test suite is provided in Section V. Section VI describes the TMS320C6416 DSP implementation. Finally the conclusions are drawn in Section VII.

2 SYSTEM DESCRIPTION

The block diagram of the proposed chaotically encrypted communication system is shown in Figure 1. The analog speech waveform is sampled and quantized using the PCM waveform coding process. The binary speech (message) signal is then masked (XORed) by the PTRBs generated using Nested Piece Wise Linear Map (NPWLM).

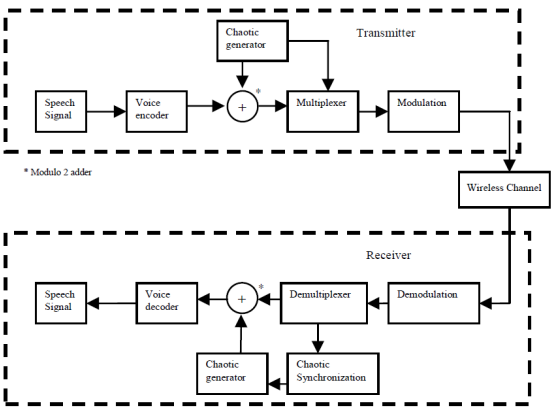

Figure 1: Block diagram of the proposed chaotic voice communication system.

As illustrated in Figure 2, the first transmitted (training) packet contains the synchronization bits produced by the PTRBG in the transmitter. The rest of the transmitted packets contain the encrypted data

Figure 2: Illustration of the synchronization bits and encrypted binary stream.

(the message signal XORed with the PTRB.

At the receiver, once the demodulated signal is received, demultixplexing is performed to extract the first packet and get the synchronization bits. The backward iteration synchronization technique is applied using the training bits which was transmitted without encryption to get an the initial condition of the chaos generator used in the transmitter. Starting the chaos generator at the receiver with the estimated initial condition, an estimated copy of the binary sequence generated at the transmitter can be produced. At this stage XORing the encrypted signal with the receiver generated chaotic binary sequence to retrieve the original message signal.

3 PSEUDO-TRUE RANDOM BIT GENERATION

A TRNG should be able to produce infinitely long sequences of independent equiprobable bits and when restarted it should never reproduce a previously delivered sequence (nonrepeatable) as it uses different initial conditions. The processes are also nondeterministic (a given sequence of numbers cannot be reproduced). Chaotic generators can be used to produce Pseudo TRBG (PTRBG) and therefore provide an efficient alternative to TRBGs. Generally speaking ideal or TRBGs are extremely difficult to implement by software. However because of the irregular behavior of chaotic systems, chaotic generators can be used to produce PTRBG and therefore provide an efficient alternative to TRBGs. Several PTRBG based on chaotic systems exist in literature, Study in this paper is focused on the nested PWLM (NPWLM).

Drutarovsky and Galajda (Drutarovsk and Galajda, 2006) proposed a modified form of the one dimensional PWLM called Nested . The map is expressed as:

$$x(n+1) = \begin{cases} 2x(n)-2 & x > 1/2 \\ 2x(n) & -1/2 < x < 1/2 \\ 2x(n)+2 & x < -1/2 \end{cases} \quad (1)$$

Equation 2 can be split and rewritten as (Drutarovsk and Galajda, 2006):

$$x'(n) = \begin{cases} x(n)-A & x(n) > 0 \\ x(n)+A & x(n) < 0 \end{cases} \quad (2)$$

$$x(n+1) = \begin{cases} B(x'(n)-A) & x'(n) > 0 \\ B(x'(n)+A) & x'(n) < 0 \end{cases} \quad (3)$$

where parameters A and B are 1.3 and 2 respectively. The domain of $x(n)$ is between -1, and $+1$. The bifurcation diagram (Abid et al., 2009), (Abid, 2009), (Abid et al., 2010) of the NPWLM shows that when the control parameter B is 2, chaos is still generated. This splitting can be thought of as a form of cascading or nesting one map into another, therefore equations (3) and (4) are called NPWLM.The PTRBGs are generated using a threshold function called the generating partition described in the following section.

4 CHAOTIC SYNCHRONIZATION

In simple terms; synchronization for discrete time systems can be thought of as the ability of the receiver to recover an identical copy of the chaotic sequence generated at the transmitter. The concept of using symbolic dynamics to process chaotic signals was studied in (Min et al., 2006), (Cong et al., 1999), where the infinite number of finite-length chaotic signals can be partitioned into a finite number of signals sets. Suppose X is the set of all possible signals for $x(n)$ that the chaotic map can generate. Using the definitions of symbolic dynamics the set X can be partitioned into M disjoint partitions E_i of the phase space S such that $\bigcup_{i=1}^{M} E_i = S$ and $E_i \bigcap E_j = \Phi$ for $i \neq j$.

In order to obtain enough information, pairs of bits are needed to be transmitted to the receiver instead of single bits. The first bit is taken from the first map and the second bit from the second map. Therefore two generating functions $b_1(n)$ and $b_2(n)$ are needed for the nested PWLM to generate the symbolic sequence given by:

$$b_1(n) = \begin{cases} 0 & x'(n) < 0 \\ 1 & x'(n) \geq 0 \end{cases} \quad b_2(n) = \begin{cases} 0 & x(n) < 0 \\ 1 & x(n) \geq 0 \end{cases} \quad (4)$$

The two bits from $b_1(n)$ and $b_2(n)$ are used to decide on the appropriate inverse chaotic map to be used, therefore a total of 38 bits are needed to retrieve the original initial condition.

Suppose that for an initial condition $x(0)$ and a length of N bits the symbolic sequence $B = [b_2(0)\,b_1(0), b_2(1)\,b_1(1), \ldots, b_2(N-1)\,b_1(N-1)]$ is generated and transmitted, where N is the number of bits required to reach an acceptable estimate of the

initial condition. The receiver use the following equation to retrieve the initial condition.

$$\widehat{x}(0|N) = f^{-1}_{b_2(0)} f^{-1}_{b_1(1)} \circ f^{-1}_{b_2(1)} f^{-1}_{b_1(1)} \circ \cdots \circ f^{-1}_{b_2(N-1)} f^{-1}_{b_1(N-1)} (\eta) \quad (5)$$

where $f^{-1}_{b(0)}$ is defined as the inverse mapping of f at bit 0, η is a point chosen randomly from the domain of $f^{-1}_{b(N-1)}(\eta)$.

When NPWLM is implemented , the following function $b(n)$ is used (what is called a generating partition) to generate the symbolic sequence.

$$b(n) = \begin{cases} 0 & x(n) < 0 \quad \text{partition } E_0 \\ 1 & x(n) \geq 0 \quad \text{partition } E_1 \end{cases} \quad (6)$$

Certainly for larger N, the estimate is more accurate. Practically, the error between the original chaotic sequence at the transmitter and the estimated sequence at the receiver must be reduced. Suppose the following system is available at the transmitter $x(n+1) = f[x(n)]$ where $x(n)$ is the chaotic value, and at the receiver the distorted value $y(n = x(n) + v(n)$ is received, where $v(n)$ is the channel noise. We target by synchronization to obtain a close estimate of x_n from the available information $y(n)$ such that $\lim_{n\to\infty} \|\widehat{x}(n) - x(n)\| = 0$ (Millerioux and Mira, 2001).

In this paper, experimental trials show that a minimum of 38 bits are needed to be sent to the receiver to achieve synchronization. The $N = 38$ bits are sufficient to reproduce the same chaotic sequence at the receiver. The number of bits required to synchronize the receiver is less than that when using chipcon platform which was $N = 50$. Fig. 2 illustrates the concept of the synchronization technique where the transmitter sends the first 38 bits unscrambled to the receiver to be used for synchronization and then the encrypted binary signal. Starting from the 38^{th} received bit and depending whether it is a zero or one; the receiver starts to iterate the appropriate inverse chaotic map starting from a random initial point η. Theoretical analysis for estimating the number of bits required for synchronization will be carried out in future work.

5 RANDOMNESS TESTS

Statistical testing is employed to provides a mechanism for making quantitative decisions that a generator produces numbers that appear to be true random. The intent is to determine whether there is enough evidence to "reject" a conjecture or hypothesis about the true randomness of the generated bits. Any random bit generator proposed for use in a cryptographic protocol must be subject to statistical tests. The sets of tests available are: a Federal Information Processing Standard (FIPS 140-1) statistical test which was lately replaced with FIPS 140-2, the German Application Notes and Interpretation of the Scheme (AIS 31) and the National Institute of Standards and Technology (NIST) test suite(NIST, 2001).

Many chaotic maps introduce biases in the binary sequence. In many cases, before testing the symbolic sequences generated by the chaotic maps, post-processing of the produced sequences has to be performed in order to reduce any biases in the produced distribution. Therefore, the well-known Von Neumann's (VN) deskewing technique can be employed. The technique consists of converting the bit pair 01 into output 0, 10 into output 1 and of discarding bit pairs 00 and 11.

Randomness tests are used to analyze the distribution pattern of the generated data. There is an array of statistical tests available to test the randomness of random and pseudorandom number generators. Even though these statistical tests do not provide definite results, it is possible to interpret these results with care and caution to determine the randomness of a generator. The general rule of thumb is more tests the better. The generator bit stream was subjected to a plenty of statistical tests for randomness used by The National Institute of Standard and Technology (NIST; an agency of the U.S. Commerce Departments Technology Administration (NIST, 2001). It is however important to note that the test suite is suitable for identifying deviations of binary sequences from randomness. However factors contributing to these deviations are numerous and it is possible to expect a certain number of failures from a particular generator. The binary sequences generated by the PWLM and NPWLM map were passed to NIST statistical suite to test their randomness. Each NIST statistical test assesses a binary sequence to establish whether there is significant evidence to suggest that the null hypothesis (H_0) should be rejected in favor of the alternative hypothesis. Here the null hypothesis H_0 is that the sequence being tested is random, while the alternative hypothesis H_1, is that the sequence being tested is not random. Thus for each applied test a decision is made to accept or reject the null hypothesis based on statistical evidence. Each test statistic obtained for each individual test is used to calculate a P-value that indicates the strength of the evidence against the null hypothesis. Thus for each test, the P-value is the probability that a perfect random number generator would have produced a sequence that is less random than the tested sequence, given the particular nonrandomness being gauged by that particular test. In this work,

the binary sequences generated by the PWLM and the NPWLM were passed to NIST statistical suite to test their randomness. They were subjected to 12 of the 16 tests of the suite. Each map was used to generate 1000 sequences each having a length of 1000000 bits. The Random Excursions, Random Excursions Variant and Non-overlapping templates tests are not applicable for these bit streams.

The DSP kit used for implementing the system definitely with finite precision; therefore the chaotic map implemented on a finite precision machine can be called a 'pseudo' chaotic map. Furthermore to determine the effect of post processing, the bit streams produced by the maps were first tested without performing any form of post processing, then the Von Neumann's deskewing technique and the XOR post processing technique were independently applied. A significance level of 0.01 was chosen for the tests. For each of the tests a P-value is calculated which is the probability that a perfect RNG would have produced a sequence less random than the sequence that was tested, given the kind of non-randomness assessed by the test. If a P-value for a test is 1, then the sequence appears to be perfectly random (Li, 2003). Table 1 summarizes the results for single precision PWLM and NPWLM. It is evident from Table 1 that the NPWLM is a good choice to be used as a PTRNG. The binary sequences generated by the NPWLM with XOR applied as a post processing technique, pass the all the tests except the FFT test.

Table 1: Summary of NIST results for TMS320C6416 based PWLM and NPWLM.

	PWLCM			NPWLM		
Type of Post Processing	None	XOR	VN	None	XOR	VN
Frequency	Pass	Fail	Pass	Pass	Pass	Pass
Block	Pass	Fail	Pass	Fail	Pass	Pass
Cumulative Sums	Pass	Fail	Pass	Pass	Pass	Pass
Cumulative Sums	Pass	Fail	Pass	Pass	Pass	Pass
Runs	Fail	Fail	Fail	Pass	Pass	Pass
Longest Runs	Pass	Fail	Fail	Pass	Pass	Fail
Rank	Fail	Fail	Fail	Pass	Pass	Pass
FFT	Fail	Fail	Fail	Fail	Fail	Fail
Overlapping	Fail	Fail	Fail	Pass	Pass	Fail
Universal	Fail	Fail	Fail	Fail	Pass	Pass
Apen	Fail	Fail	Fail	Pass	Pass	Fail
Serial	Fail	Fail	Fail	Pass	Pass	Fail
Serial	Fail	Fail	Fail	Pass	Pass	Fail
Linear Complexity	Pass	Pass	Fail	Pass	Pass	Pass

Figure 3: Implementation setup of the TMS320C6416 development kit.

6 SYSTEM IMPLEMENTATION

The proposed system is implemented on fixed-point TMS320C6416 DSP manufactured by Texas Instruments Corporation (TI) (TexasInstruments, 2011).

The generation and synchronization code is written in C language and compiled using Code Composer Studio Workbench Software. The program starts by generating the first 38 chaos bits using the NPWLM and sending them without encryption. This is followed by further chaotic generation and XORing of the symbolic sequence with the digital data samples. Then the required data transmission of the encrypted packets is carried out. The receiver part of the program receives the encrypted data and performs synchronization bits by estimating the initial value using the backward iteration synchronization technique to regenerate an exact copy of the chaotic symbolic sequence generated at the transmitter. At this point the encrypted data in the buffer can be decrypted to retrieve the original message and play it with the kit DAC.

Real-Time Data Exchange (RTDX) is used to provide real time, continuous visibility into the way TRBG software application operate in TMS320C6416. RTDX allows transfer the random bits generated in the DSP to a host PC for testing. On the host platform, an RTDX host library operates in conjunction with Code Composer Studio. In RTDX an output channel should be configured within NPWLM code which resides on the DSP kit. The generated data from NPWLM is written to the output channel. This data is immediately recorded into a C6416 DSP buffer defined in the RTDX C6416 library. The data from this buffer is then sent to the host PC through the JTAG interface. The RTDX host library receives this data from the JTAG interface and records it into either a memory buffer for testing purposes.

7 CONCLUSIONS

Application of the backward iteration synchronization method for linear chaotic nested maps was introduced and implemented. Pseudo-true random bits generated the PWLM, and NPWLM using double and single precisions test bed (DSP chip) were tested. The test results have shown that the bits generated by the NPWLM with XOR applied as a post processing technique and using a single precision representation of numbers, pass the maximum number (11 out of 12) of tests for 1000 binary sequences each having a length of 1000000 bits. The chaotic map was implemented on TMS320C6416 DSP development kit. The application of the backward iteration synchronization method to nested maps required the estimate of the initial condition from two concatenated maps as compared to one for normal chaotic map. After extended experimental tests, the results proved that a minimum of 38 synchronization chaotic bits are needed to achieve synchronization.

REFERENCES

Abid, A., Nasir, Q., and Elwakil, A. S. (2009). Implementation of a chaotically encrypted wireless communication system. In *International Conference on Communications 2009 (ICC'09), June 14-18,, 2009, Dresden Germany*.

Abid, A., Nasir, Q., and Elwakil, A. S. (2010). Implementation of an encrypted wireless communication system using nested chaotic maps. In *International Journal of Bifurcation and Chaos, DOI: 1O.1142/S0218127410027957, In press*.

Abid, A. M. (2009). An implementation of chaotic encryption in wireless voice transmission. In *M.Sc. Thesis, Electrical and Computer Engineering, University of Sharjah, Sharjah, UAE, Feb 2009*.

Ahmed, J. and Siyal, M. Y. (2005). Fft based analog speech scrambler using tms320c6711 dsp. In *9th International Multitopic Conference, IEEE INMIC, pp. 1 - 4, Dec. 2005*.

Ahmed, J. and Siyal, M. Y. (2006). A robust secure speech communication system using itu-t g.723.1 and tms320c6711 dsp. In *Microprocessor and Microsystems, vol. 30, no. 1, pp. 26-32, Feb 2006.*

Cong, L., Xiaofu, W., and Songgeng, S. (1999). A general efficient method for chaotic signal estimation. In *TEMPLATE'06, 1st International Conference on Template Production*.

De Angeli, A., Genesio, R., and Tesis, A. (1995). Dead-beat chaos synchronization in discrete-time systems. In *IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications, vol. 42, no. 1, pp. 54-56, Jan 1995.*

Drutarovsk, M. and Galajda, P. (2006). Chaos based true random number generator embedded in a mixed-signal reconfigurable hardware. In *Journal of Electrical Engineering, Vol. 57, pp. 218-225, April 2006.*

Lau, Y. (2006). Techniques in secure chaos communication. In *Ph.D. Thesis, School of Electrical and Computer Engineering Science, RMIT University, Victoria, Australia, Feb. 2006*.

Li, S. (2003). Analyses and new designs of digital chaotic ciphers. In *Ph.D. dissertation, School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, June 2003.*

Masuda, N. and Aihara, K. (1999). Cryptosystems with discretized chaotic maps. In *IEEE Transactions on Circuits and Systems, Vol. 49, pp. 28-40, January 1999*.

Matthews, R. (1989). On the derivation of a chaotic encryption. In *Cryptologia XIII (1), Vol. 8, pp. 29-49, January 1989.*

Millerioux, G. and Mira, C. (2001). Finite-time global chaos synchronization for piecewise linear maps. In *IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications, vol. 48, pp. 111-116, Jan 2001.*

Min, L., Fei, P., and Shuisheng, Q. (2006). Implementation of a new chaotic encryption system and synchronization. In *Journal of Systems Engineering and Electronic, vol. 17, no. 1, pp. 43-47, 2006.*

NIST (2001). A statistical test suite for random and pseudorandom number generators for cryptographic applications. In *NIST 800-22, May 2001*.

Sajeeth, N., Philip, K., and Babu, J. (2001). Chaos for stream cipher. In *ADCOM 2000*.

Sprott, J. C. (2003). *Chaos and time-series analysis*. Oxford University Press.

Stojanovski, T. and Kocarev, L. (1997). pplications of symbolic dynamics in chaos synchronization. In *IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications, vol. 44, no. 10, pp. 1014-1018, Oct. 1997.*

Tang, K. W. and Tang, W. K. S. (2005). Chaos-based secure voice communication system. In *IEEE International Conference on Industrial Technology, ICIT, pp. 571-576, Dec 2005.*

Tao, S., Ruili, W., and Yixun, Y. (1999). The theoretical design for a class of new chaotic feedback stream ciphers. In *Acta Eletronica Sinica, Vol. 27, pp. 47-50, July 1999.*

TexasInstruments (2011). Tms320c6416-dsp. http://focus.ti.com/docs/prod/folders/print/tms320c6416.html.

Yong-Ai, Z. and Yi-Bei, N. (2002). Impulsive synchronization of discrete chaotic systems. In *Chinese Physical Letters, vol. 20, no. 2, pp. 199-201, Sep. 2002.*