Telecommun Syst
DOI 10.1007/s11235-011-9460-1

A chaotic encryption scheme for real-time embedded systems: design and implementation

Amit Pande · Joseph Zambreno

Abstract Chaotic encryption schemes are believed to provide greater level of security than conventional ciphers. In this paper, a chaotic stream cipher is first constructed and then its hardware implementation details over Xilinx Virtex-6 FPGA are provided. Logistic map is the simplest chaotic system and has high potential to be used to design a stream cipher for real-time embedded systems. Its simple construct and non-linear dynamics makes it a common choice for such applications. In this paper, we present a Modified Logistic Map (MLM) which improves the performance of Logistic Map in terms of higher Lyapunov exponent and uniformity of bifurcation map. It also avoids the stable orbits of logistic map giving a more chaotic behavior to the system. A stream cipher is built using MLM and random feedback scheme. The proposed cipher gives 16 bits of encrypted data per clock cycle. The hardware implementation results over Xilinx Virtex-6 FPGA give a synthesis clock frequency of 93 MHz and a throughput of 1.5 Gbps while using 16 hardware multipliers. This makes the cipher suitable for embedded devices which have tight constraints on power consumption, hardware resources and real-time parameters.

Keywords Chaos · Encryption · Stream cipher · FPGA implementation

A. Pande (✉) · J. Zambreno
Department of Electrical and Computer Engineering, Iowa State University, Ames, USA
e-mail: amit@iastate.edu

J. Zambreno
e-mail: zambreno@iastate.edu

Published online: 02 June 2011

1 Introduction

Chaos theory plays an active role in modern cryptography. As the basis for developing a crypto-system, the advantage of using chaos lies in its random behavior and sensitivity to initial conditions and parameter settings to fulfill the classic Shannon requirements of confusion and diffusion [24]. To meet a great demand for real-time secure image transmission over the Internet, a variety of encryption schemes have been proposed. Of them, chaos-based algorithms have shown some exceptionally good properties in many concerned aspects regarding security, complexity, speed, computing power and computational overhead, etc.

Chaotic systems are characterized by sensitive dependence on initial conditions, similarity to random behavior, and continuous broad-band power spectrum. The possibility for self-synchronization of chaotic oscillations [21] has sparked an avalanche of works on application of chaos in cryptography. The random behavior and sensitivity to initial conditions and parameter settings allows chaotic systems to fulfill the classic Shannon requirements of confusion and diffusion [24]. A tiny difference in the starting state and parameter setting of these systems can lead to enormous differences in the final state of the system over a few iterations. Thus, sensitivity to initial conditions manifests itself as an exponential growth of error and the behavior of system appears chaotic.

Several schemes have been developed which allow transforming the information signal into a chaotic waveform on the transmitter side and to extract the information signal from the transmitted waveform on the receiver side. The most important among them are: chaotic masking, chaos shift keying, and chaotic modulation.

A lot of research in mathematics and communications has been devoted to study of continuous-time chaotic systems such as the oscillator circuits [5, 6, 29]. However, these schemes need a synchronization procedure. In this work, we focus on discrete-time chaotic systems which behave like private-key encryption algorithms [23] and are amenable to implementation in fixed point hardware.

Many chaotic block ciphers [2, 9, 13, 22, 31] have been proposed in research literature. They can be broadly divided into two types: chaotic block ciphers and chaotic stream ciphers. The work by Baptista [2] was one of the earliest attempts to build a block cipher based on chaotic encryption. The basic idea is to encrypt each character of the message as the integer number of iterations performed in the logistic equation, in order to transfer the trajectory from an initial condition towards a pre-defined interval inside the logistic chaotic attractor. However, there are some limitations of block ciphers proposed using chaotic maps.

Firstly, the distribution of the ciphertext is not flat enough to ensure high security since the occurrence probability of cipher blocks decays exponentially as the number of iterations increases. Secondly, the encryption speed of these cryptographic schemes is very slow since at least 250 iterations of the chaotic map are required for encrypting an 8-bit symbol. The number may vary upto 65532. Thirdly, the length of ciphertext is at least twice that of plaintext, a byte of message may result in several tens of thousands of iterations that need two bytes to carry. Although papers showing some improvements in encryption speed, and cipher text size have been proposed, block ciphers remain slow to suffice the encryption needs of real-time data (and multimedia) encryption systems.

Chaotic encryption has also been used to design image encryption schemes. [10] present a four dimensional chaotic cipher for secure image transmission using Arnold 2-D chaotic maps. [4] present a compression and encryption scheme that uses a variable and unpredictable statistical model for arithmetic coding generated using pseudo-random bitstream generated by a couple of chaotic systems. However, many schemes have proven to be weak against cryptanalysis using known-plaintext attacks and others [1, 3].

Other implementations of chaos for secure data communication are found in [29] (security enhancement via delta modulation; [28] (integrating chaotic encryption with arithmetic coding); [5] (Differential delayed feedback) etc.

A stream cipher based on chaotic map was presented in early 1991 by [25] and its cryptanalysis was presented by [3]. Chen et al. [9, 31] constructed a block cipher based on three-dimensional maps while [22] proposed a cipher by direct discretization of two dimensional Baker map. A good survey and introductory tutorial on these schemes is found in [11, 30]. [16] present a crypto-system based on a discretization of the skew tent map. [17] presents chaotic Feistel and chaotic uniform operations for block ciphers. Although various schemes/maps have been proposed in research literature, the logistic map remains one of the most simplest map and used in many schemes.

In this paper we present the design and implementation of a chaotic stream cipher that uses less hardware, has promising security and has high throughput to serve the requirements of real-time embedded systems. The main contributions of this paper can be summarized as under:

1. We present a Modified Logistic Map which has better properties than the Logistic Map—in terms of higher confusion (larger Lyapunov exponent) and a flatter distribution for various parameter values in the bifurcation diagram.
2. This paper gives a overview of existing chaotic ciphers, and presents a new stream cipher which can resist the known attacks by simple modifications in the encryption algorithm.
3. To the best knowledge of the authors, this is the first hardware implementation of a chaotic stream cipher in hardware.
4. We present an optimized implementation of 64 bits multiplication in FPGA leading to savings in hardware resource requirements.
5. A throughput of 1.5 Gbps was obtained for Virtex-6 XCVLX75TL FPGA. The design was synthesized and implemented using Xilinx ISE 11.0 tool.

The paper is organized as follows. Section 2 gives a brief overview of existing research and the desired properties of a good stream cipher. Section 3 gives details of the stream cipher algorithm. In Sect. 4, we discuss the properties of MLM and resistance of proposed cipher against cryptanalysis. Section 5 gives the details of hardware implementation over Xilinx Virtex-6 FPGA while Sect. 6 concludes the paper with directions of future work.

2 Stream cipher

Many different chaotic systems have been employed to generate pseudo-random keystream, 2-D Henon attractor in [7], logistic map in [15], generalized logistic map in [12, 14, 18, 20], quasi-chaotic nonlinear filter in [8], and piecewise linear chaotic map in [32]. Several chaotic stream ciphers [8, 18] have been known to be insecure [1, 26] and known plaintext attacks have been proposed.

Besides, cryptographic security some of the factors influencing the design of a good chaotic stream cipher for real-time applications are as follows:

1. **Finite range of control parameter**: Logistic map has been widely investigated in chaos theory and is very simple to be realized, hence it has been used by many digital chaotic ciphers [15, 18]. However, only when the control parameter λ_{LM} (defined formally in a later section) is 4.0, logistic map is a surjective function and has perfect chaotic properties. λ_{LM} must be selected near 4.0

in these ciphers, which makes the key space very small. Moreover, some unsuitable λ_{LM} values are also found in vicinity of $\lambda_{LM} = 4$ making the range and choice of key-space narrow and constrained. Piecewise linear chaotic maps, such as tent map [32] and others [8] can be used but their piecewise linearity may lead to overall weakness of cryptosystem [3].

2. **Encryption Speed and hardware implementation**: Some digital chaotic ciphers are too slow to be feasible for real-time encryption [25]. While the chaotic systems are running in finite precision, the fixed-point arithmetic is preferable over floating point mathematics which require more hardware resources and computation time. However, several chaotic systems are defined by some complicated functions [18] and must run under floating-point arithmetic. They should be avoided in chaotic ciphers. Adding multiple chaotic maps increases the security of cryptosystem but also increases the hardware resources (and power consumption) required in implementation which may not be feasible for low power embedded systems scenario. Some ciphers [2] have time-variant speed, so they cannot encrypt plaintext with constant bit-rate, such as MPEG video stream.
3. The dynamics of discrete chaotic systems is different than those for the continuous-time chaotic systems. Discretization leads to severe degradations such as short cycle-length, non-ideal distribution and correlation, etc. These issues need to be properly addressed in the design of a cryptosystem.

3 Algorithmic description

In this subsection, we give a brief description of the proposed algorithm. The proposed scheme is robust to the choice of initial conditions (due to lack of any unsuitable λ values) achieves real-time encryption speed and has desirable properties of a chaotic cryptosystem.

3.1 The pseudo-random number generator based on a Modified Logistic Map

3.1.1 The Logistic Map

The Logistic Map is a polynomial mapping of degree 2. It demonstrates chaotic behavior although using a simple non-linear dynamical equation. Mathematically, the logistic map is written as

$$x_{n+1} = \lambda_{LM} \times x_n(1 - x_n)$$

where λ_{LM} is a positive number.

The behavior of logistic map is dependent on the value of λ_{LM}. At $\lambda_{LM} \approx 3.57$ is the onset of chaos, at the end of the period-doubling cascade. We can no longer see any oscillations. Slight variations in the initial population yield dramatically different results over time, a prime characteristic of chaos. Most values beyond 3.57 exhibit a chaotic behavior, but certain isolated values of λ_{LM} appear to show non-chaotic behavior and are called as islands of stability. Beyond $\lambda_{LM} = 4$, the values eventually leave the interval $[0, 1]$ and diverge for almost all initial values.

A rough description of chaos is that chaotic systems exhibit a great sensitivity to initial conditions—a property of the logistic map for most values of λ between about 3.57 and 4. This stretching-and-folding does not just produce a gradual divergence of the sequences of iterates, but an exponential divergence, evidenced also by the complexity and unpredictability of the chaotic logistic map.

3.1.2 The Modified Logistic Map (MLM)

Our initial experimentation involved generation of pseudo-random number sequences by varying the parameter λ_{LM} in the range $[3.57, 4]$. It led to several observations:

1. The histogram obtained for different λ_{LM} values (with 50000 samples) is skewed and not uniform or flat. This is illustrated for $\lambda_{LM} = 3.61$ and $\lambda_{LM} = 3.91$ values in Fig. 1(a, b). The distribution for $\lambda_{LM} = 4$ is most flat and symmetric (see Fig. 1(c)). It is desirable to have a flatter distribution of samples drawn from the logistic map in order to increase its randomness.
2. For $\lambda_{LM} = 4$, the logistic map equation $x_{n+1} = \lambda_{LM} \times x_n(1 - x_n)$ has the same domain and range intervals $(0, 1)$. For $\lambda_{LM} < 4$ and input x_n in range $(0, 1)$, the range of x_{n+1} in the expression is $(0, \lambda_{LM}/4]$ and the distribution of random numbers is biased towards 0 or 1 (as seen in distributions in Fig. 1(a, b)). It is desirable to have a distribution of random numbers symmetric around 0.5.
3. There are certain isolated values of λ_{LM} that appear to show non-chaotic behavior and are called as islands of stability. For example: $\lambda_{LM} = 1 + \sqrt(8) \approx 3.83$ show oscillation between three values.
4. $\lambda_{LM} = 4.0$ has most flat, uniform and symmetric histogram than other λ_{LM} values.

We address these issues by developing a MLM, defined by the following equation:

$$x_{n+1} = \lambda \times x_n(1 - x_n) + \mu$$

where the x_n values are restricted to the interval $[\alpha, 1 - \alpha]$, $\alpha < 0.5$. The maxima of this function occurs at $x_n = 0.5$ and the maximum value is $\lambda/4 + \mu$ while the minimum (in specified domain) occurs at $x_n = \alpha$ or $x_n = 1 - \alpha$ and the minimum value is $\lambda \times \alpha(1 - \alpha) + \mu$. Equating the maximum

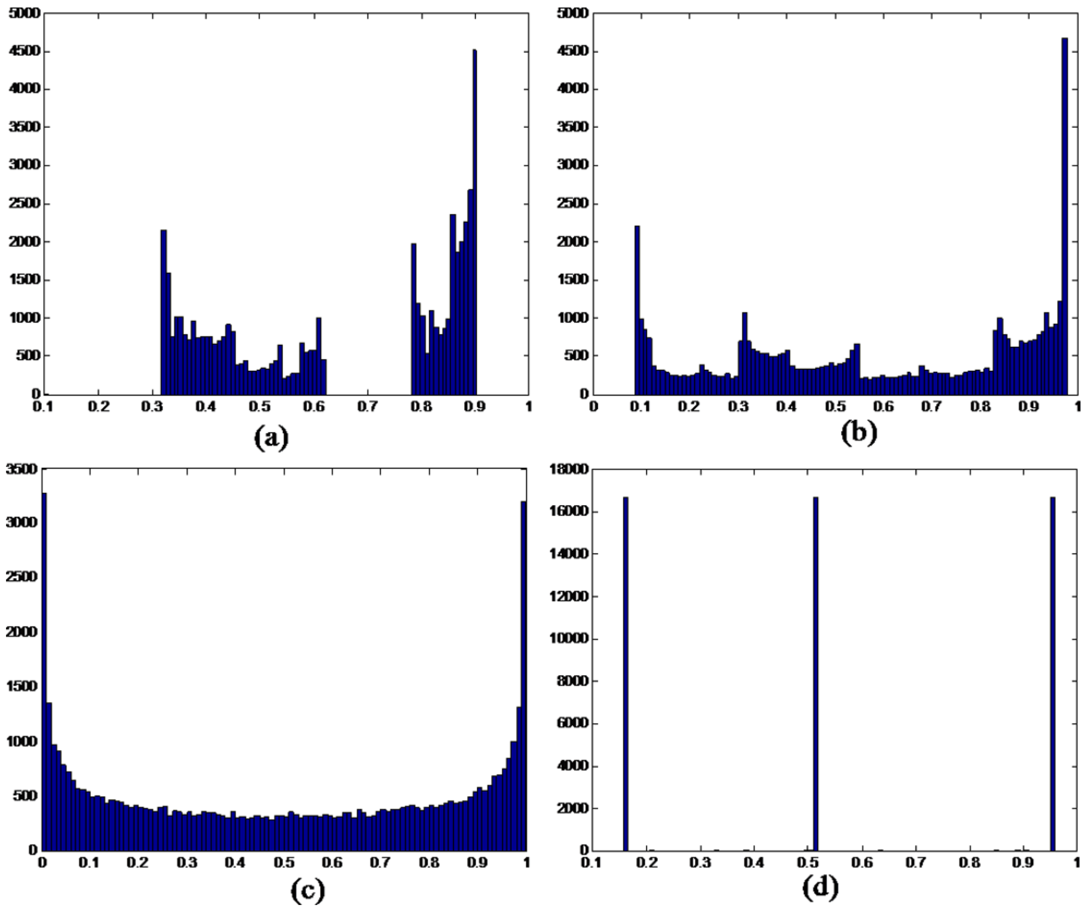

Fig. 1 Histogram for 50000 samples obtained using Logistic map with initial seed 0.100010 and (**a**) $\lambda_{LM} = 3.61$ and (**b**) $\lambda_{LM} = 3.91$ (**c**) $\lambda_{LM} = 4$ and (**d**) $\lambda_{LM} = 3.83$

and minimum values to the range $[\alpha, (1-\alpha)]$ leads to the following equations:

$$\alpha = \lambda\alpha(1-\alpha) + \mu$$

$$1 - \alpha = \frac{\lambda}{4} + \mu$$

On solving these equations, we get $\lambda = \frac{4}{1-2\alpha}$ and $\mu = \frac{\alpha(2\alpha-3)}{1-2\alpha}$. Substituting these values, we get a flatter histogram for the new logistic map as evident in Fig. 2. This modified logistic map addresses the requirements of flatter and symmetric distribution and also avoids islands of stability by generating a flat distribution for all values of α.

3.2 Quantization

The output of the modified logistic map (x_n) is quantized to get a 16 bit value y_n. $x_n, 0 < x_n < 1$ is represented in fixed point as follows:

$$x_n = \sum_{j=0}^{N-1} \{a_j\} \times 2^{j-N}$$

where a_j are individual bit values.

We target the logistic map implementation on N bits hardware architectures but restrict y_n to the least significant 16 bits only. Thus, y_n is given by:

$$y_n = \sum_{j=0}^{15} \{a_j\} \times 2^{j-N}$$

The quantization step or truncation of more significant bits is non-linear in nature (it is a many-one mathematical function) thereby increasing the complexity of any attacks that try to recover the logistic map information from the cipher text using any cryptanalysis. We extract another single bit from the logistic map output which is used later for the random feedback scheme. For example, the single bit output sequence b_n can be obtained from the bits of x_n as follows:

$$b_n = \{a_{N-1}\}$$

i.e. the MSB of x_n is used to get b_n.

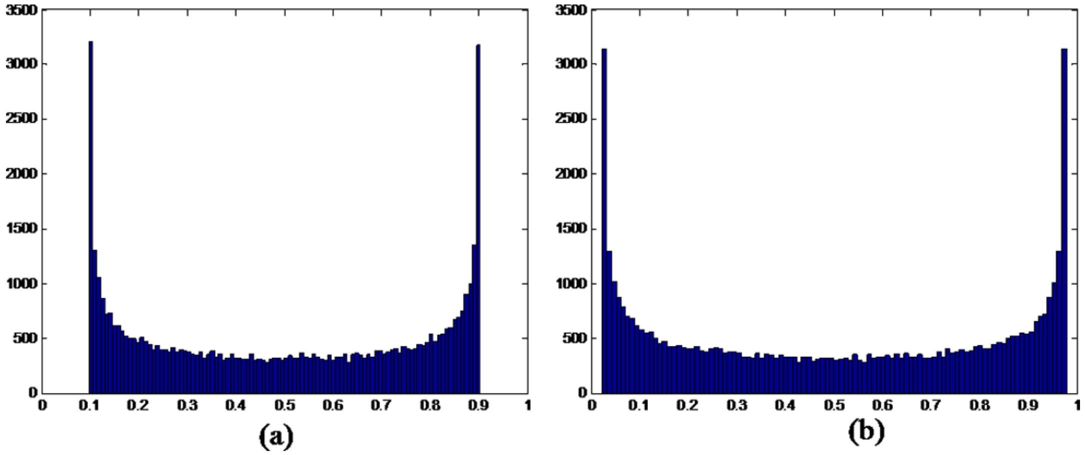

Fig. 2 Histogram for 50000 samples obtained using Modified Logistic Map with α values corresponding to (**a**) $\lambda_{LM} = 3.61$ and (**b**) $\lambda_{LM} = 3.91$

3.3 Pseudo-random sequence-2

We generate another pseudo-random sequence z_n from the given sequence y_n by the following operation:

$$z_n = y_n \oplus y_{n-1} \oplus y_{n-2}$$

There is no linear correlation between the two sequences y_n and z_n. Statistical de-correlation makes it difficult to backtrack y_n from z_n.

3.4 Masking operation and random feedback

The ciphertext C_n is obtained from the plaintext P_n by the following operation:

$$C_n = P_n \oplus z_n \oplus Fb_n$$

where z_n is the pseudorandom sequence and Fb_n is the random feedback input from the past ciphertext output. The value Fb_n is obtained as follows:

$$Fb_n = \begin{Bmatrix} C_{n-1} \text{ when } b_n = 0 \\ C_{n-2} \text{ when } b_n = 1 \end{Bmatrix}$$

4 Resistance against cryptanalysis

The performance and accuracy of discrete chaotic ciphers is a translation of properties of the underlying dynamical system (or chaotic map). The chaotic properties of logistic maps and hence MLM have been established in the past decades by several researchers [19].

Shannon [24] explains that a good crypto-system must show diffusion and confusion properties. Confusion refers to making the relationship between the key and the ciphertext as complex and involved as possible while diffusion means that the output bits should depend on the input bits in a very complex way i.e. a change in a bit in input plain text should imply a change in output bit with a probability of $\frac{1}{2}$. Chaotic systems show random behavior and inherently exhibit confusion with respect to the initial conditions (x_0) and the parameter (α) that make the key. We perform some statistical tests to test the pseudo-random nature of the key obtained.

4.1 Randomness tests

We perform the following randomness tests to study the pseudo-random nature of sequence (b_n) generated using the proposed scheme.

4.1.1 Frequency test

In a randomly generated N-bit sequence we would expect approximately half the bits in the sequence to be ones and approximately half to be zeroes. The frequency test checks that the number of ones in the sequence is not significantly different from $N/2$.

Based on 1000 simulations on strings of length 10000 each generated using variable initial values and control parameter, the probability for zero and one were obtained to be 0.4993 and 0.5007 respectively for the sequence b_n. For the non-binary sequence z_n, frequency test was performed by discretizing the sequence around its mean value. We observed the probability of zeros and one in this sequence to be 0.4981 for 1000 simulations of length 10000.

4.1.2 Serial test

The serial test checks that the frequencies of the different transitions in a binary sequence (i.e., 11, 10, 01, and 00) are approximately equal. This will then give us an indication as to whether or not the bits in the sequence are independent of their predecessors.

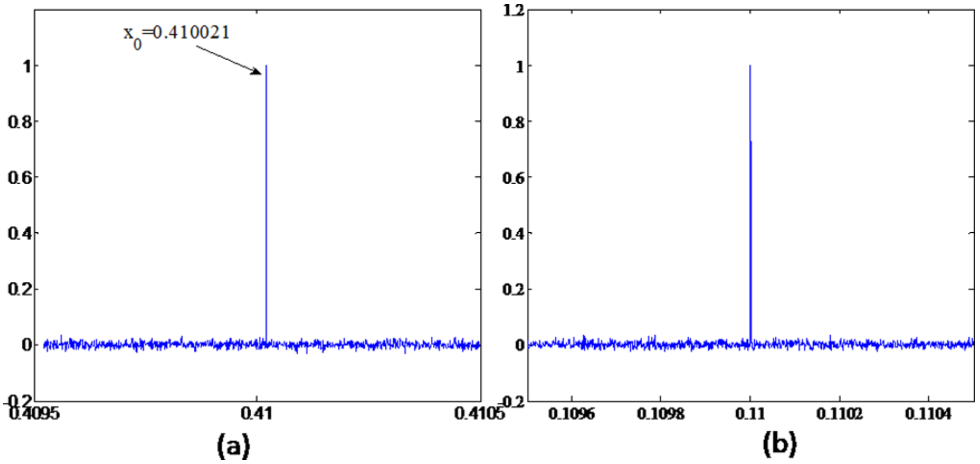

Fig. 3 Correlation test of the pseudo-random sequence. (**a**) Generated using different initial values x_0 and (**b**) different initial parameter α. The plots are measured against initial value $\alpha = 0.110000$ and $x_0 = 0.410021$

For the sequence b_n, 1000 simulations of 10000 samples were run. The probabilities for getting 00, 01, 10 and 11 were found to be 0.2503, 0.2491, 0.2480, and 0.2526 respectively (the ideal distribution would give 0.25 for all probabilities).

4.1.3 *Runs test*

The binary sequence is divided into blocks (runs of ones) and gaps (runs of zeroes). The runs test checks that the number of runs of various lengths in our sequence are similar to what we would expect to find in a random sequence. This test is only applied if the sequence has already passed the serial test in which case it is known that the number of blocks and gaps are in acceptable limits.

This is a test of the hypothesis that the values in a sequence come in a random order, against the alternative that the ordering is not random. For non-binary sequences (such as z_n) the test is based on the number of runs of consecutive values above or below the mean of input sequence. Too few runs is an indication of tendency of high values to cluster together, and low values to cluster together. Too many runs is an indication of a tendency for high values and low values to alternate. Tests were performed using Matlab simulations. The result is $H = 0$ if the null hypothesis ("sequence is random") cannot be rejected at the 5% significance level, or $H = 1$ if the null hypothesis can be rejected at the 5% level. We ran 10000 simulations with different initial values and parameter settings, giving us 8916 successful simulations with $H = 0$.

4.1.4 *Statistical properties*

Some of the necessary conditions for a secure stream cipher are long period, large linear complexity, randomness and proper order of correlation immunity [23]. A long period is assured by taking a large value of N (say 64). Figure 3(a) and (b) show the low correlation between sequences obtained using slightly different (a) initial value x_0 and (b) parameter λ. It can be seen (see Table 1) that a very poor correlation is obtained amongst sequences generated using slightly different initial condition or parameter.

Table 1 Statistical performance of generated sequence b_n (results based on 1000 sequences of length 10000 each)

Probabilities of Zero	0.4993
Probabilities of One	0.5007

4.2 Bifurcation map

If the dynamical system under consideration is a chaotic map, then the orbit derived from any initial condition covers the whole phase space. This is seen with the help of bifurcation diagram of logistic maps. A bifurcation diagram is the plot of sample set of x_n obtained against the variations in initial parameter λ_{LM}.

The bifurcation map of logistic map is shown in Fig. 4(a). It is observed that for some value of λ_{LM}, the logistic map reaches a few stable states and oscillate around them. These regions must be removed carefully from the key space. Hence, an exhaustive elimination of stable points (corresponding to white spaces in bifurcation diagram) is necessary to build a scheme based on Logistic Map.

Figure 4(b) shows the bifurcation map of MLM as a function of free parameter α. It can be seen that there are no free white spaces in the bifurcation diagram, indicating no in-between regions of stable oscillations in MLM. Thus, the entire range of parameter α can be used to build the key space.

4.3 Lyapunov exponent

Lyapunov exponent is a measure of stability of non-linear systems. It characterizes the rate of separation of infinitesimally close trajectories. The maximum Lyapunov exponent

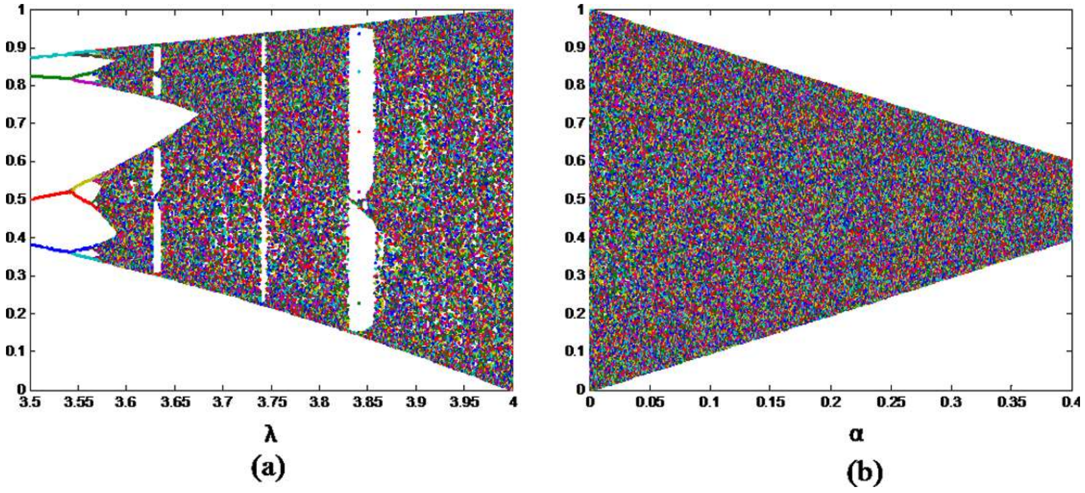

Fig. 4 Bifurcation Diagram for (**a**) Logistic Map showing the white spaces (islands of stability) and asymmetricity and (**b**) Modified Logistic Map with symmetric and flatter distribution

is defined by the following expression:

$$\Lambda = \lim_{t\to\infty} \frac{1}{t} \ln \frac{|\delta Z(t)|}{|\delta Z_0|}$$

where $\delta Z(t)$ is the separation at time t and δZ_0 is the initial divergence. In our cipher, if we choose two different initial values x_{0a} and x_{0b}, which are very close to each other such that $x_{0a} - x_{0b} \approx \delta Z_0$, a positive Lyapunov exponent will indicate that the two trajectories will diverge from each other. The discrete time equivalent expression to find Lyapunov exponent of MLM will be:

$$\begin{aligned}\Lambda &= \lim_{n\to\infty} \frac{1}{n} \ln \frac{|\delta x_n|}{|\delta x_0|} \\ &= \lim_{n\to\infty} \frac{1}{n} \ln \frac{|\delta x_n|}{|\delta x_{n-1}|} \frac{|\delta x_{n-1}|}{|\delta x_{n-2}|} \cdots \frac{|\delta x_1|}{|\delta x_0|}\end{aligned}$$

An analysis similar to logistic map [27] can be performed to prove the positive Lyapunov exponent for logistic maps.

$$x_n = \lambda \times x_{n-1}(1 - x_{n-1}) + \mu$$

Hence,

$$\left|\frac{\delta x_n}{\delta x_{n-1}}\right| = |\lambda \times (1 - 2x_{n-1})|$$

Therefore, we can express Λ as follows:

$$\begin{aligned}\Lambda &= \lim_{n\to\infty} \frac{1}{n} \left(\sum_{j=1}^{j=n} \ln \left| \frac{\delta x_j}{\delta x_{j-1}} \right| \right) \\ &= \lim_{n\to\infty} \frac{1}{n} \left(\sum_{j=1}^{j=n} \ln \left| \lambda(1 - 2x_j) \right| \right)\end{aligned}$$

The value of Λ can be calculated by running a numerical trial of large number of samples (say 10,000) starting with any randomly picked initial value x_0. The values of Lyapunov exponent for Logistic Map and MLM are plotted in Fig. 5(a) and (b). This value was found to be $\ln 2$ for MLM which is the same as the value for Logistic Map with $\lambda_{LM} = 4$. Thus, the divergence rate of MLM, measured by Lyapunov coefficient is always greater than or equal to the value for Logistic Map. This indicates better confusion properties of MLM. Moreover, it is independent of α indicating the invariance of confusion properties with the change in parameter α.

4.4 General security of the scheme

A serious drawback of chaotic cryptosystems is that they are weak against known-plaintext attacks. If the plain-text and the cipher-text are known, it is easy to XOR both the values and obtain the key value that was XORed to the original plaintext. Our proposed scheme lays many practical difficulties against such reverse engineering:

- The random feedback scheme makes it difficult to predict the key value XORed to the original plaintext.
- The sequences z_n and y_n are linearly uncorrelated from each other making it difficult to reverse engineer the values of y_n from z_n.
- The sequence y_n is obtained by sampling of x_n which is used to iterate the chaotic map. In the hardware implementation (presented in next section), we sample the Least Significant 16 bits (out of 64) of x_n to get y_n. Because, the chaotic map is more sensitive to the MSB than to the LSB (and we have 48 unknown MSB bits), it is practically impossible to trace back the x_n value.

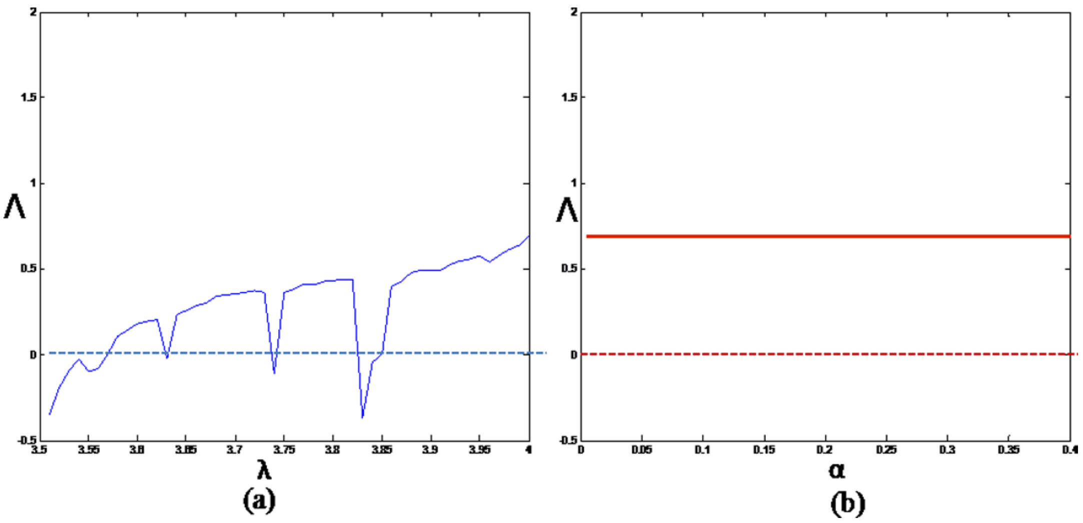

Fig. 5 Plot of Lyapunov Coefficient (Λ *solid line*) for (**a**) Logistic Map as a function of parameter λ_{LM} indicating regions of non-chaotic behavior and (**b**) Modified Logistic Map showing higher divergence than Logistic Map and independence of Λ from parameter α

- We allowed 100 iterations of MLM in the beginning to allow the diffusion of initial key bits and parameter values. It was found that within approximately 20 iterations of Logistic Map the initial parameter values are fully diffused: the two logistic maps with a slight difference in initial conditions will appear completely de-correlated in their outputs after at most 20 iterations. Allowing 100 iterations, help us to be on a safer side to allow full diffusion of the initial key parameters.

Thus, the presented scheme is secure against known-plaintext attacks. In the next section we present a hardware implementation of the scheme that uses 128 bit encryption key (64 bits each for initial condition and parameter λ settings).

5 Hardware implementation

For hardware implementation, we chose a fixed point implementation over floating point implementation because fixed point operations can be implemented more efficiently in hardware. The bit width of the Plaintext and the Ciphertext are 16 bits or 2 bytes. However, for the implementation of MLM, we chose a bit width of 64 bits. Thus, the iterating value of MLM ($x(i)$ and the parameters λ and μ are both implemented with 64 bits floating point precision).

The permissible range of parameter α was chosen to be $(0, 0.375)$ which is represented in fixed point with 0 integer bits and 64 fractional bits. This is represented shortly as 0.64 in I.F (Integer.Floating point) format. The range for parameter λ is then calculated to be $(4, 16)$ which is implemented with 5.59 I.F format. The range for μ is $(-3, -15.0975)$ which is represented using 5.59 I.F format. Thus, the multiplication $\lambda \times x(i) \times (1 - x(i))$ is truncated to 5.59 I.F format and then added to μ to obtain the new value for $x(i)$.

The parameter α can take 3×2^{61} values while the parameter x_0 can take approximately 2^{63} values. Thus, we get an effective keyspace with 3×2^{124} or approximately 2^{125} key values to choose from.

We synthesized the design over a Xilinx Virtex-6 XCVLX75TL FPGA using Xilinx ISE 11.0. The new XtremeDSP DSP48E1 slice in Virtex-6 SXT series facilitates faster and optimized DSP functions (including multiplications). They can deliver over 1 TeraMACs at 550 MHz with up to 2016 user-configurable XtremeDSP DSP48E1 slices and cuts the power consumption by 65% using innovative, efficient power management. A direct implementation of the design gave a clock frequency of 35 MHz. By adding two pipelining stages to the multiplier (DSP48E1 slices), we get a clock frequency of 70 MHz for the design.

A single DSP48E1 slice can perform a maximum of 25×18 bits multiplication and hence 12 slices are required for a 64×64 bits multiplication. Two multiplication require 24 DSP48E1 slices. However, since we truncate the 128 bit output of 64×64 bits multiplication to only 64 bits, some optimization is possible. Xilinx XST (Synthesis tool) thus reduces one DSP48E slice by optimization thus requiring 23 slices for implementation.

We present an optimization of usage of DSP multipliers based on above observations for the multiplication of two 64 bit numbers X and Y. X is sign extended to 72 bits (X_{SE} and represented by $X_a X_b X_c$ where X_a,X_b and X_c are each 24 bit long sequences.

$$\{X_{SE}\}_0^{71} = \{X_a\}_{48}^{71}\{X_b\}_{24}^{47}\{X_c\}_0^{23}$$

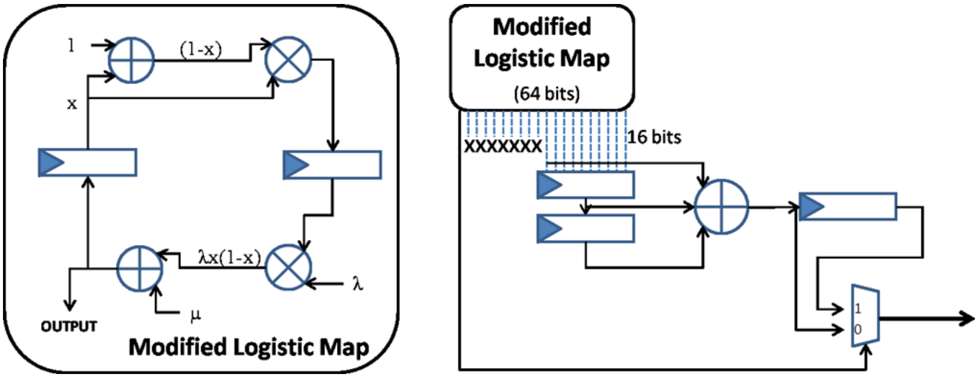

Fig. 6 Block diagram showing the implementational details of the chaotic stream cipher

Similarly, we can represent Y as combination of four 16 bit numbers $Y_w Y_x Y_y Y_z$.

$$\{Y\}_0^{63} = \{Y_w\}_{48}^{63}\{Y_x\}_{32}^{47}\{Y_y\}_{17}^{31}\{Y_z\}_0^{15}$$

Numerically,

$$X = X_{SE} = X_a \times 2^{48} + X_b \times 2^{24} + X_c$$

and

$$Y = Y_w \times 2^{48} + Y_x \times 2^{32} + Y_y \times 2^{16} + Y_z$$

The product $X \times Y$ can then be represented as:

$$\begin{aligned} X \times Y &= (X_a \times 2^{48} + X_b \times 2^{24} + X_c) \times (Y_w \times 2^{48} \\ &\quad + Y_x \times 2^{32} + Y_y \times 2^{16} + Y_z) \\ \Rightarrow X \times Y &= 2^{96} \times X_a Y_w + 2^{72} \times X_b Y_w + 2^{48} \times X_c Y_w \\ &\quad + 2^{80} \times X_a Y_x + 2^{56} \times X_b Y_x + 2^{32} \times X_c Y_x \\ &\quad + 2^{64} \times X_a Y_y + 2^{40} \times X_b Y_y + 2^{16} \times X_c Y_y \\ &\quad + 2^{48} \times X_a Y_z + 2^{24} \times X_b Y_z + 2^{0} \times X_c Y_z \end{aligned}$$

Now, considering the product $X_n(1 - X_n)$ in the logistic map, we multiply two 0.64 I.F values to get an output which is in 0.128 I.F format. We truncate the last 64 bits to get the 64 bit approximate value of X_{n+1}. Because X is represented in 72 bits, we can discard lower 72 bits of the product. Each of the product $X_\alpha Y_\beta$, such that $\alpha \in \{a, b, c\}$ and $\beta \in \{w, x, y, z\}$ is of size 40 bits and can be implemented in a single DSP48E1 slice.

Thus,

$$\begin{aligned} X \times Y = 2^{96} &\times X_a Y_w + 2^{72} \times X_b Y_w + 2^{48} \times X_c Y_w \\ &+ 2^{80} \times X_a Y_x + 2^{56} \times X_b Y_x \\ &+ 2^{64} \times X_a Y_y + 2^{40} \times X_b Y_y \\ &+ 2^{48} \times X_a Y_z \end{aligned}$$

Table 2 Resource Utilization on Xilinx Virtex-6 FPGA

	Orig. Design	Opt. Design
Clock Frequency (MHz)	69	93
No. DSP48E1 slices	23	16
No. Slice Registers	228	160
No. Slice LUTs	354	643

The other multiplication operation can also be optimized in a similar manner. Thus, we can reduce the hardware requirements and critical path for the implementation. A direct implementation of our scheme using the above optimization achieved a clock frequency of 44 MHz on the above mentioned FPGA. By adding two pipelining stages to the 64 × 64 bits multiplier, we obtained a clock frequency of 93 MHz and required only 16 DSP48E1 slices in the design. The design summary are given in Table 2. Further pipelining may lead to higher clock frequency but also increase slice registers usage.

Figure 6 gives the block diagram of the hardware implementation of the encryption scheme. MLM based PRNG (Pseudo-Random Number Generator) is shown in dotted thin lines. Blue thick lines indicate the pipelining stages. The input x_{n-1} is first multiplied with $(1 - x_{n-1})$ and the upper half bits (most significant bits) are then multiplied with λ. The output of this multiplication is then truncated and added with μ to get the value of x_n as shown in the figure. The output x_n is also used to extract the values y_n and b_n, both of which serve to generate the output cipher text. As shown in the figure, the multiplexer (mux) is used to provide the random feedback based on bit b_n.

6 Discussion

In this paper, a chaotic stream cipher is designed using logistic maps and its hardware implementation over a Virtex-6 FPGA has been presented. Chaotic ciphers are simpler to design, have excellent mathematical properties and are becoming popular for data encryption, image encryption and other schemes.

6.1 Comparison to stream ciphers

Classical stream ciphers are generally constructed using Linear Feedback Shift Registers (LFSR). The main disadvantage of LFSR based structure is its vulnerability to attack due to inherent linearity in the structure. Stream ciphers built over LFSRs introduce non-linearity into the design either using a suitable cryptographic Boolean function or irregular clocking.

On the other hand,the chaotic stream cipher presented in this paper has the advantages of random behavior and dependence to initial conditions. To the best knowledge of the authors, it is the first implementation of chaos-based PRNG on FPGA. The proposed implementation (with hardware optimizations as proposed earlier) achieves a throughput of 1.5 Gbps over Virtex-6 FPGA. By pipelining the proposed architecture, we obtain a clock frequency of 350 MHz corresponding to a throughput of 5.6 Gbps.

The existing stream ciphers such as A5/1, A5/2 have been attacked in real-time. Attack has been mounted on RC4, PANAMA, FISH etc (not in real-time) but they use a very large internal state (and thus require large more hardware resources). On the contrary, our scheme has an internal representation of just 64 bits (against 2064 bits for RC4, 1216 bits for PANAMA) and is resistant to known cryptanalysis.

6.2 Comparison to existing chaotic ciphers

The implementation presented in this paper uses fixed point arithmetic as against floating point arithmetic presented in [18] which requires implementation of complicated functions. Unlike the chaotic cipher presented by [2] which needs arbitrary number of cycles for each encode and very large number of iterations (upto 65532), the presented cipher has constant (1) iteration of logistic map per encode and gives a throughput as high as 5.6 Gbps.

[8, 32] use piecewise linear maps to build a chaotic cipher which have been reported to be weak against cryptanalysis. On the other hand, the ciphers based on logistic map are non-linear in nature [12, 14, 18, 20] and have several weaknesses. There are islands of stability in bifurcation maps of logistic maps. In this work, we use modified logistic map which removes islands of stability and makes the distribution more uniform as discussed earlier. The weaknesses of logistic maps against reverse engineering schemes are addressed in this paper as follows:

1. Using only 16 of the 64 bit output from iterations on logistic map.
2. XORing with time-delayed outputs
3. Non-linear random feedback

These three details have been explained in Sect. 3.

These considerations make our scheme a suitable candidate for end-end encryption in real time embedded systems such as mobile phones, portable video and audio players, and cameras.

7 Conclusion

This paper presents a novel chaotic stream cipher based on modified logistic map suitable for embedded real-time applications. A hardware implementation of proposed scheme was proposed and a clock frequency of 93 MHz was achieved.

A possible direction for future work is the study of more complex chaotic maps, study of behaviors of coupled-chaotic maps and their implementation over hardware platforms.

References

1. Alivarez, G., Montoya, F., Romera, M., & Pastor, G. (2004). Cryptanalysis of dynamic look-up table based chaotic cryptosystems. *Physics Letters A*, *326*(3–4), 211–218. doi:10.1109/81.974872.
2. Baptista, M. S. (1998). Cryptography with chaos. *Physics Letters*, *240*(1–2), 50–54.
3. Biham, E. (1991). Cryptanalysis of the chaotic-map cryptosystem suggested at eurocrypt'91. In *Advances in cryptology in EUROCRYPT 91. Lecture notes in computer science* (pp. 532–534). Berlin: Springer.
4. Bose, R., & Pathak, S. (2006). A novel compression and encryption scheme using variable model arithmetic coding and coupled chaotic system. *IEEE Transactions on Circuits and Systems I, Fundamental Theory and Applications*, *53*(4), 848–857. doi:10.1109/TCSI.2005.859617.
5. Robilliard, C., & Huntington, J. W. E. H. (2006). Enhancing the security of delayed differential chaotic systems with programmable feedback. *IEEE Transactions on Circuits and Systems II, Express Briefs*, *53*(8), 722–726. doi:10.1109/TCSII.2006.876405.
6. Carroll, T. L. P. L. (1991). Synchronizing chaotic circuits. *IEEE Transactions on Circuits and Systems*, *38*(4), 453–456. doi:10.1109/31.75404.
7. Forre, R. (1991). The henon attractor as a keystream generator. In *Advances in cryptology EUROCRYPT 91. Lecture notes in computer science* (pp. 76–81). Berlin: Springer.
8. Frey, D. (1993). Chaotic digital encoding: an approach to secure communication. *IEEE Transactions on Circuits and Systems II, Express Briefs*, *40*(10), 660–666. doi:10.1109/82.246168.
9. Chen, G., Mao, C. K. C. Y. (2004). A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons and Fractals*, *21*(3), 749–761.
10. Hamdi, M., & Boudriga, N. (2008). Four dimensional chaotic ciphers for secure image transmission. In *IEEE intl. conf. multimedia and expo* (pp. 437–440). doi:10.1109/ICME.2008.4607465.
11. Kocarev, L. (2001). Chaos-based cryptography: a brief overview. *IEEE Circuits and Systems Magazine*, *1*(3), 6–21. doi:10.1109/7384.963463.

12. Kurian, A. P. P. S. (2008). Self-synchronizing chaotic stream ciphers. *Signal Processing*, *88*(10), 2442–2452. doi:10.1016/j.sigpro.2008.04.003.
13. Kocarev, L., Jakimoski, G., Stojanovski, T., & Parlitz, U. (1998). From chaotic maps to encryption schemes. In *Proceedings of the 1998 IEEE international symposium on circuits and systems* (Vol. 4, pp. 514–517). New York: IEEE Press.
14. Liu, S. X. Z. C. Z., & Jing, S. (2008). An improved chaos-based stream cipher algorithm and its vlsi implementation. In *Intl. conf. networked computing and advanced information management* (pp. 191–197).
15. Bianco, M. E. D. A. R. (1991). Encryption system based on chaos theory. US Patent No. 5,048,086.
16. Masuda, N., & Aihara, K. (2002). Cryptosystems with discretized chaotic maps. *IEEE Transactions on Circuits and Systems I, Fundamental Theory and Applications*, *49*(1), 28–40. doi:10.1109/81.974872.
17. Masuda, N. A. K. K. L., Jakimoski, G. (2006). Chaotic block ciphers: from theory to practical algorithms. *IEEE Transactions on Circuits and Systems I, Fundamental Theory and Applications*, *53*(6), 1341–1352. doi:10.1109/TCSI.2006.874182.
18. Matthews, R. (1989). *Cryptologia*, *XIII*(1), 29–42.
19. May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, *261*, 459–467.
20. Philip, N. S. K. J. (2000). Chaos for stream cipher. In *Proc. recent adv. computing communications, ADCOM2000* (pp. 35–42). New York: Tata McGraw-Hill.
21. Pecora, L. M., & Carroll, T. (1990). Synchronization in chaotic systems. *Physical Review Letters*, *64*(8), 821–824.
22. Pichler, F., & Scharinger, J. (1996). Finite dimensional generalized baker dynamical systems for cryptographic applications. In *EUROCAST '95: select. papers fifth intl. work. computer aided systems theory* (pp. 465–476). London: Springer.
23. Rueppel, R. (1986). *Analysis and design of stream ciphers*. Berlin: Springer.
24. Shannon, C. E. (1949). Communication theory of secrecy systems. *The Bell System Technical Journal*, *28*, 656–715.
25. Habutsu, T. I. S. S. M., & Nishio, Y. (1991). A secret key cryptosystem by iterating a chaotic map. In *Advances in cryptology EUROCRYPT 91. Lecture notes in computer science* (pp. 127–140). Berlin: Springer.
26. Wheeler, D. D. (1991). Problems with chaotic cryptosystems. *Cryptologia*, *XV*(2), 140–151.
27. Wolf, A. (1986). *Quantifying chaos with Lyapunov exponents*. Princeton: Princeton University Press.
28. Wong, K. W., & Yuen, C. H. (2008). Embedding compression in chaos-based cryptography. *IEEE Transactions on Circuits and Systems II, Express Briefs*, *55*(11), 1193–1197. doi:10.1109/TCSII.2008.2002565.
29. Liang, X., & Zhang, J. (2008). Improving the security of chaotic synchronization with a delta-modulated cryptographic technique. *IEEE Transactions on Circuits and Systems II, Express Briefs*. *55*(7), 680–684. doi:10.1109/TCSII.2008.921585.
30. Yang, T. (2004). A survey of chaotic secure communication systems. *International Journal of Computational Cognition*, *2*, (2).
31. Mao, Y. G. C., & Lian, S. (2004). A symmetric image encryption scheme based on 3d chaotic baker maps. *International Journal of Bifurcation and Chaos*, *14*(10), 3613–3624.
32. Zhou, H. L. X. (1997). Generating chaotic secure sequences with desired statistical properties and high security. *International Journal of Bifurcation and Chaos*, *7*(1), 205–213.

Amit Pande is a graduate student working with Dr. Joseph Zambreno on his dissertation 'Algorithms and Architectures for Secure Embedded Multimedia Systems'. He completed his Bachelors in Electronics and Communications Engineering from IIT Roorkee, India where his major project was awarded with Institute Silver Medal. The same project also won third place at 'India Innovation Initiative' (then called as Agilent Engiineering and Technology Award) 2007. His research interests are in multimedia security, embedded systems, security and image processing.

Joseph Zambreno received the B.S. degree (Hons.) in computer engineering in 2001, the M.S. degree in electrical and computer engineering in 2002, and the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, IL, in 2006. Currently, he is an Assistant Professor in the Departmentof Electrical and Computer Engineering at Iowa State University, Ames, where he has been since 2006. His research interests include computerarchitecture, compilers, embedded systems, andhardware/ software co-design, with a focus on run-time reconfigurable architectures and compiler techniques for software protection. Dr. Zambreno was a recipient of a National Science Foundation GraduateResearch Fellowship, a Northwestern University Graduate School Fellowship, a Walter P. Murphy Fellowship, and the Electrical Engineering and Computer Science Department Best Dissertation Award for his Ph.D. dissertation "Compiler and Architectural Approaches to SoftwareProtection and Security."