Reconfigurable hardware implementation of a modified chaotic filter bank scheme

A. Pande* and J. Zambreno

Department of Electrical and Computer Engineering,
Iowa State University,
Ames, IA-50011, USA
Fax: 515-294-1152
E-mail: amit@iastate.edu
E-mail: zambreno@iastate.edu
*Corresponding author

Abstract: Chaotic filter bank schemes have been proposed in the research literature to allow for the efficient encryption of data for real-time embedded systems. Some security flaws have been found in the underlying approaches which makes such a scheme unsafe for application in real life scenarios. In this paper, we first present an improved scheme to alleviate the weaknesses of the chaotic filter bank scheme, and add enhanced security features, to form a modified chaotic filter bank (MCFB) scheme. Next, we present a reconfigurable hardware implementation of the MCFB scheme. Implementation on reconfigurable hardware speeds up the performance of MCFB scheme by mapping some of the multipliers in design to reconfigurable look-up tables, while removing many unnecessary multipliers. An optimised implementation on Xilinx Virtex-5 XC5VLX330 FPGA gave a speedup of 30% over non-optimised direct implementation. A clock frequency of 88 MHz was obtained.

Keywords: chaos; encryption; stream cipher; FPGA implementation.

Reference to this paper should be made as follows: Pande, A. and Zambreno, J. (2010) 'Reconfigurable hardware implementation of a modified chaotic filter bank scheme', *Int. J. Embedded Systems*, Vol. 4, Nos. 3/4, pp.248–258.

Biographical notes: Amit Pande is currently a graduate student at Iowa State University, USA, pursuing his research in developing efficient reconfigurable hardware architectures to enable multimedia security and compression. He was the 3rd winner of the design contest at the VLSI Design Conference 2009. He completed his graduation from IIT Roorkee in Electronics and Communications Engineering where the major project was awarded Institute Silver Medal, and third best major project in Agilent Engineering and Technology Awards (India) 2007. His research interests are in multimedia compression, security, e-learning, and reconfigurable computing.

Joseph Zambreno has been with the Department of Electrical and Computer Engineering at Iowa State University since 2006, where he is currently an Assistant Professor. Prior to joining ISU, he was at Northwestern University in Evanston, IL, where he graduated with his PhD in Electrical and Computer Engineering in 2006, MS in Electrical and Computer Engineering in 2002, and BS Summa Cum Laude in Computer Engineering in 2001. While at Northwestern University, He was a recipient of a National Science Foundation Graduate Research Fellowship, a Northwestern University Graduate School Fellowship, a Walter P. Murphy Fellowship, and the EECS Department Best Dissertation Award for his PhD dissertation titled 'Compiler and architectural approaches to software protection and security'.

1 Introduction

1.1 Chaos and cryptography

Chaos theory plays an active role in modern cryptography. As the basis for developing a cryptosystem, the advantage of using chaos lies in its random behaviour and sensitivity to initial conditions and parameter settings to fulfil the classical Shannon requirements of confusion and diffusion (Shannon, 1949). A tiny difference in the starting state and parameter setting of these systems can lead to completely different outputs over a few iterations. Thus, sensitivity to initial conditions manifests itself as an exponential growth of error and the behaviour of system appears chaotic.

Quite a bit of research has been devoted to the study of continuous-time chaotic systems such as the oscillator circuits (Carroll and Pecora, 1991; Liang et al., 2008; Robilliard et al., 2006). However, these schemes need a synchronisation procedure. On the other hand, discrete-time chaotic systems behave like private-key encryption

algorithms (Rueppel, 1986) and are amenable to implementation on fixed point hardware.

Many chaotic block ciphers (Baptista, 1998; Kocarev et al., 1998; Guanrong Chen and Chui, 2004; Yaobin Mao and Lian, 2004; Pichler and Scharinger, 1996) have been proposed in research literature. For example, Baptista (1998) builds a block cipher based on chaotic encryption. Each character of the message is encoded 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.

Some limitations of such block ciphers and the logistic chaotic attractor are explained as follows:

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 of iterations may vary up to 65,532. Thirdly, the length of ciphertext is at least twice that of plaintext, X bits of message may result in several tens of thousands of iterations that need $2X$ bytes to carry. Despite the improvements proposed by subsequent research, block ciphers based on Baptista's (1998) work remain slow to satisfy the encryption needs of the real-time data encryption systems.

A stream cipher was designed over chaotic maps and presented in early 1991 by Habutsu et al. (1991). Its cryptanalysis was presented in the same conference (Biham, 1991). Guanrong Chen and Chui (2004), and Yaobin Mao and Lian (2004) constructed a block cipher based on three-dimensional maps while Pichler and Scharinger (1996) proposed a cipher by direct discretisation of two dimensional Baker map. A good survey and introductory tutorial on these schemes is found in Yang (2004) and Kocarev (2001). The authors in Masuda and Aihara (2002) present a crypto-system based on a discretisation of the skew tent map. Masuda et al. (2006) presents chaotic Feistel and chaotic uniform operations for block ciphers. Although various schemes/maps have been proposed in the research literature, the logistic map remains one of the simplest maps and is used in many schemes.

1.2 Wavelets and chaotic filter banks

Chaotic filter banks based cipher was proposed by Ling et al. (2007). It allows great flexibility in the design and gives the following advantages:

1 One can embed signals in different frequency bands by employing different chaotic functions.

2 The number of chaotic generators to be employed and their corresponding functions can be selected and designed in a flexible manner because perfect reconstruction does not depend on the invertibility, causality, linearity and time invariance of the corresponding chaotic functions.

3 The ratios of the subband signal powers to the chaotic subband signal powers can be easily changed by the designers and perfect reconstruction is still guaranteed no matter how small these ratios are.

4 The proposed cryptographic system can be easily adapted to the international multimedia standards, such as JPEG 2000 and MPEG-4 (Ling et al., 2007).

The encryption procedure is carried out by decomposing the input plaintext signal into two different subbands and masking each of them with a pseudorandom number sequence generated by iterating the chaotic logistic map. The authors (Ling et al., 2007) use the discrete wavelet transform (DWT) based filters banks in their approach to maintain compatibility with existing image compression standards such as JPEG 2000 (Christopoulos et al., 2000).

Arroyo et al. (2009) presents a cryptanalysis of Ling et al. (2007) which exposes weaknesses of chaotic filter bank against known plaintext attacks and also exposes the limitation of reduction of key space by use of logistic map.

1.3 Scope and organisation of this paper

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 summarised as follows:

1 The proposed modified chaotic filter bank (MCFB) scheme is a lightweight cipher designed to satisfy the resource requirements of real-time embedded systems, security requirements of modern communication systems and format-compliance with existing multimedia compression standards such as JPEG 2000, MPEG-4, etc.

2 To the best of knowledge of the authors, this is the first hardware implementation of a chaotic filter bank scheme in hardware.

3 A clock frequency of 88 MHz was obtained for a Virtex-5 XC5VLX330 FPGA. The design was synthesised and implemented using Xilinx ISE 10.1 tool.

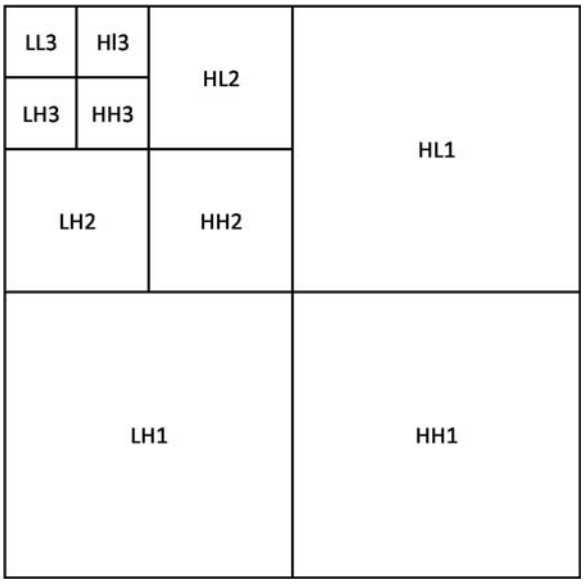
The paper is organised as follows: Section 2 gives a brief overview of the wavelet transform. Section 3 gives details of the chaotic filter bank scheme proposed earlier. In Section 4, we discuss the MCFB scheme and subsequently discuss its distinguishing features in Section 5 and Section 6. Section 5 explains the improved chaotic oscillator (ICO) and Section 6 gives an overview of wavelet parameterisation. Section 8 gives the details of hardware implementation over Xilinx Virtex-5 FPGA and the

proposed optimisations, while Section 9 concludes the paper with directions of future work.

2 Wavelets

The efficient representation of time-frequency information by the wavelet transform has led to its popularity for signal processing applications. It provides superior rate-distortion and subjective image quality performance over existing standards. Applying a 2D DWT to an image of resolution $M \times N$ results in four images of dimensions $\frac{M}{2} \times \frac{N}{2}$: three are detailed images along the horizontal (LH), vertical (HL) and diagonal (HH), and one is coarse approximation (LL) of the original image. LL represents the low frequency component of the image, while LH, HL, and HH represent the high frequency components. This LL image can be further decomposed by DWT operation. Three levels of such transforms are applied and shown in Figure 1. The coarse information is preserved in the LL3 image and this operation forms the basis of multi-resolution analysis for DWT (Vetterli and Kovačevic, 1995).

Figure 1 Result of three level 2D wavelet transform operation on an image

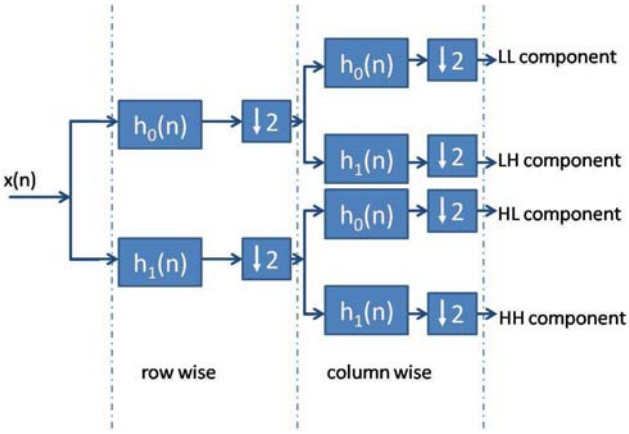


Prior works in signal processing explain that the 1D DWT can be viewed as a signal decomposition using specific low pass and high pass filters. A single stage of image decomposition can be implemented by successive horizontal row and vertical column wavelet transforms. Thus, one level of DWT operation is represented by filtering with high and low pass filters across row and column successively and is explained in Figure 2. After each filtering a down sampling is done by a factor of 2 to remove the redundant information.

2.1 Commonly used DWT filters

The two most common DWT filters used in image compression are Gall's 5/3 filter and the Daubechies 9/7 filter (Christopoulos et al., 2000). They are accepted in the JPEG 2000 standards. The Gall's filter has rational coefficients and its hardware implementation requires less resources. The Daubechies 9/7 (also commonly known as CDF 9/7) filter has better compression performance. However, it has irrational coefficients therefore its hardware requirements are very large.

Figure 2 Basic stages of a one level 2D wavelet transform operation (see online version for colours)



2.1.1 Daubechies 9/7-tap bi-orthogonal filter

The biorthogonal Daubechies 9/7 filter is the most widely used filter for DWT operation. These wavelets have symmetric scaling and wavelet functions, i.e., both the low pass and high pass filters are symmetric. This filter has excellent image compression capabilities. There are four filters that comprise the two-channel biorthogonal wavelet system. The analysis and synthesis low-pass filters are denoted by H_0 and G_0 respectively. The analysis and synthesis high pass filters are denoted by H_1 and G_1 respectively and are obtained by quadrature mirroring the low-pass filters.

$$H_1(z) = z^{-1}G_0(-z), G_1(z) = zH_0(-z) \tag{1}$$

If we define $D(z) = G_0(z)H_0(z)$ the perfect reconstruction (PR) condition simplifies to the following:

$$D(z) + D(-z) = 2 \tag{2}$$

This equation is solved using Lagrange half band filters (LHBF), $L_K(z)$ where :

$$D(z) = L_K(z) = z^K \left(\frac{1+z^{-1}}{2}\right)^{2K} \alpha(k), \tag{3}$$

$$\alpha(k) = \sum_{n=0}^{K-1} \left({}^{K+n-1}C_n\right) \left(\frac{2-\left(z+z^{-1}\right)}{4}\right)^n \tag{4}$$

This is simplified for $K = 4$ to get the famous Cohen-Daubechies-Feauveau filter also known as Daubechies biorthogonal 9/7 filter. The filter coefficients are irrational.

2.1.2 Gall's 5/3 filter

Gall and Tabatabai (1988) solved the PR condition by substituting $D(z) = a_0 + a_2z^{-2} + a_3z^{-3} + a_2z^{-4} + a_0z^{-6}$ with the condition $a_0 \in \left[-\frac{1}{8}, 0\right]$. For $a = \frac{1}{16}$ the simplification leads to the famous Gall's 5/3 filter pair. This filter has lower latency than the ones studied earlier but provides lesser image compression capabilities.

2.2 Reconfigurable hardware implementation

Much research has been done in the development of DWT architectures for image processing (Benkrid et al., 2001, 2003; Ritter and Molitor, 2001; Kotteri et al., 2005; Martina and Masera, 2007). A good survey on architectures on DWT coding is given by Tseng et al. (2005).

Recent works in partial reconfiguration of FPGAs implement DWT in a reconfigurable fashion. Claus et al. (2008) gives a comparison of embedded reconfigurable video-processing architectures. They propose a hybrid of two hardware platforms: one providing easy reconfiguration of modules and the other providing easy implementation with higher clock frequency, to achieve an optimal FPGA-based dynamically and partially reconfigurable platform for real-time video and image processing. The tool ReCoBus-Builder (Koch et al., 2008) simplifies the generation of dynamically reconfigurable systems to almost a push button process. The work also describes a communication infrastructure for dynamically reconfigurable systems.

3 Chaotic filter bank scheme

The chaotic filter bank scheme is illustrated in Figure 3. A chaotic function $\alpha_i()$ is used to create chaotic response to the system.

$$\alpha_i(n) = n + s_i(n), \quad i \in \{1, 2\}$$

where $s_i(n)$ is the output of chaotic map.

The various signals in Figure 3 are expressed as follows:

$$y_0[n] = \sum_{\forall m} x[m]h_0[2n - m],$$

$$y_1[n] = \sum_{\forall m} x[m]h_1[2n - m],$$

$$z_0[n] = y_0[n] + \alpha_0\left(y_1[n]\right),$$

$$\text{and } z_1[n] = y_1[n] + \alpha_1\left(y_0[n]\right),$$

$$\Rightarrow z_0[n] = y_0[n] + y_1[n] + s_0[n],$$

$$\text{and } z_1[n] = y_1[n] + y_0[n] + s_1[n],$$

The reconstructed signal $x'[n]$ must be the same as the original signal $x[n]$. At the decoder, first the effect of mixing with chaotic signals is reversed and then corresponding inverse wavelet transform is applied.

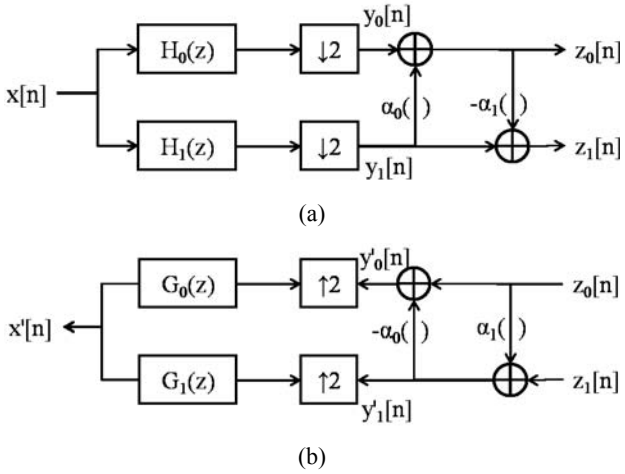
$$y_1'[n] = z_1[n] + \alpha_1\left(z_0[n]\right),$$

$$y_0'[n] = z_0[n] - \alpha_0\left(z_1[n]\right),$$

$$x'[n] = \sum_{\forall m} y_0'[m]g_0[n - 2m] + \sum_{\forall m} y_1'[m]g_1[n - 2m]$$

where h_0, h_1 are so-called analysis and g_0, g_1 are synthesis filters. Choosing Gall's 5/3 filter or Daubechies 9/7 filters allow correct recovery of the plain text signal.

Figure 3 Block diagram representation of the chaotic filter bank scheme (a) the encryption module and (b) the decryption module



3.1 Chaotic maps

As explained above, the chaotic filter bank scheme uses two chaotic maps $\alpha_0()$ and $\alpha_1()$ for its operation. These chaotic maps are based on the logistic map.

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

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

where λ_{LM} is a positive number.

The behaviour 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 behaviour, but certain isolated values of λ_{LM} appear to show non-chaotic behaviour 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.

Table 1 Coefficients for the CDF 9/7 filter

i	$h_0(i)$	$h_1(i)$
±4	0.026748757411	0
±3	–0.016864118443	0.091271763114
±2	–0.078223266529	–0.057543526229
±1	0.266864118443	–0.591271763114
0	0.602949018236	1.11508705
i	$g_0(i)$	$g_1(i)$
±4	0	0.026748757411
±3	–0.091271763114	0.016864118443
±2	–0.057543526229	–0.078223266529
±1	0.591271763114	–0.266864118443
0	1.11508705	0.602949018236

3.2 Key space

The authors in Ling et al. (2007) suggest using the initial values of logistic map and the value of parameter λ_{LM} to build the key space.

Arroyo et al. (2009) present a cryptanalysis of the above mentioned scheme and exposes some weaknesses of the scheme. They are enumerated as follows:

1 Reduction of the key space (Ling et al., 2007) proposes to use the entire range [3, 4] as the key space. The values of λ_{LM} in the interval [3, 3.57)] do not produce any chaos. Besides this, there are many points (known as islands as islands of singularity) in the interval [3.57.4] where iteration on logistic map leads to oscillation among finite values [see Figure 4(d)]. Another issue is the non-uniform distribution of output values [as shown in Figure 4(a)–Figure 4(b)].

2 Vulnerability to known plain-text attack. The value of λ_{LM} can be calculated very accurately from two successive iterations of the logistic map leading to successful plain text attacks on the scheme.

4 The MCFB scheme

The MCFB scheme makes three modifications to the original scheme, making it more secure and also improving its frequency resolution.

1 The chaotic filter bank scheme (Ling et al., 2007) involves mixing of low pass and high pass coefficients. This mixing hampers the compression performance of the wavelet transform. The equations for $z_0[n]$ and $z_1[n]$ have $y_1[n]$, and $y_0[n]$ terms in expressions for $z_0[n]$ and $z_1[n]$ respectively which lead to loss of frequency resolution of DWT.

   The new relationship between $z_0[n]$ and $z_1[n]$ is given by the following equations:

   $$z_0[n] = y_0[n] + s_0[n],$$

   and

   $$z_1[n] = y_1[n] + s_1[n]$$

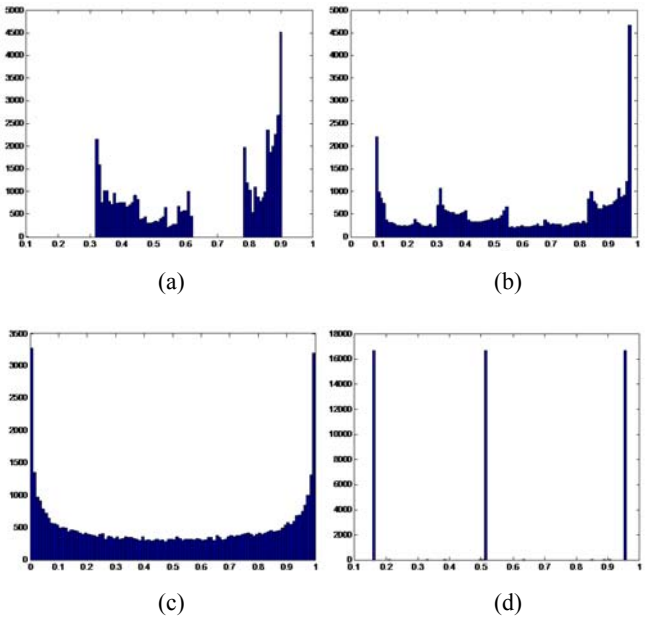
2 We use an ICO instead of the standard logistic map. This chaotic oscillator, although derived from the standard logistic map, is strong against known cryptanalysis of logistic map-based ciphers and chaotic filter banks. Moreover, it has a large continuous key space as against logistic map which has very limited key space with regions of stability within the same range.

3 We replace the DWT filter banks with a parameterised filter bank that yields has the same properties as the original filters but allows us to choose from a very large number of possible filters while implementing a filter bank.

   The choice of filter bank and parameters for the chaotic oscillators used in the design is governed by a key. The overall system is shown in Figure 5.

The ICO and parameterised wavelet transform are explained in following two sections.

Figure 4 Histogram for 50,000 samples obtained using Logistic map with initial seed 0.100010 and (a) $\lambda_{LM} = 3.61$ (b) $\lambda_{LM} = 3.91$ (c) $\lambda_{LM} = 4$ (d) $\lambda_{LM} = 3.83$ (see online version for colours)



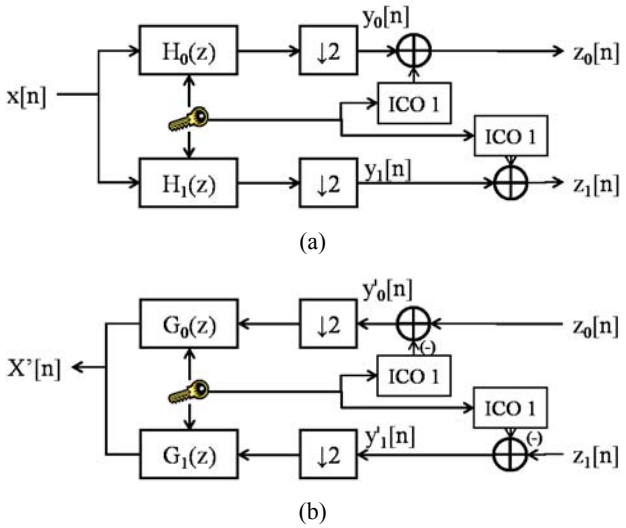
(a) (b)

(c) (d)

5 Improved chaotic oscillator

In this subsection, we give a brief description of an ICO, based on a MLM that alleviates the problems associated with chaotic generator proposed in Ling et al. (2007). 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 resistant to known attacks.

Figure 5 Block diagram representation of the MCFB scheme (a) the encryption module and (b) the decryption module (see online version for colours)



5.1 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 50,000 samples) is skewed and not uniform or flat. This is illustrated for $\lambda_{LM} = 3.61$ and $\lambda_{LM} = 3.91$ values in Figure 4(a)–Figure 4(b). The distribution for $\lambda_{LM} = 4$ is most flat and symmetric [see Figure 4(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 xn+1 in the expression is (0, λ_{LM} /4] and the distribution of random numbers is biased towards 0 or 1 [as seen in distributions in Figure 4(a)–Figure 4(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 behaviour 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:

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

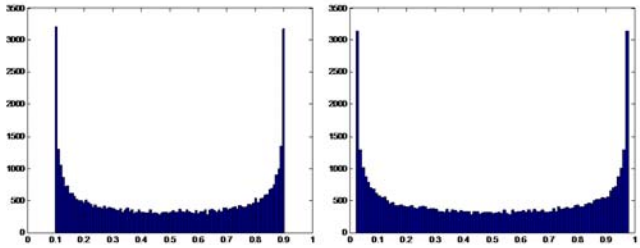
where the x_n values are restricted to the interval [α, 1 − α], $\alpha < 0.5$. The maxima of this function occurs at $x_n = 0.5$ and the maximum value is $\lambda/4 + \mu$ while the minimum (in specified domain) occurs at $x_n = \alpha$ or $x_n = 1 - \alpha$ and the minimum value is $\lambda \times (1 - \alpha) + \mu$. Equating the maximum and minimum values to the range [α, (1 − α)] 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 Figure 6. This MLM addresses the requirements of flatter and symmetric distribution and also avoids islands of stability by generating a flat distribution for all values of α.

Figure 6 Histogram for 50,000 samples obtained using modified logistic map with α values corresponding to (a) $\lambda_{LM} = 3.61$ and (b) $\lambda_{LM} = 3.91$ (see online version for colours)



(a) (b)

The output of the MLM (x_n) is quantised to get a 16 bit value p_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.

Thus, p_n is given by:

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

The quantisation 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 generate another pseudo-random sequence s_n from the given sequence p_n by the following operation:

$$s_n = p_n \oplus p_{n-1} \oplus p_{n-2}$$

There is no linear correlation between the two sequences p_n and s_n. Statistical de-correlation makes it difficult to back-track p_n from s_n.

6 Wavelet parameterisation

We now present a new layout and configuration scheme for the parameterised DWT. A new parameterised construction of the DWT filter with rational coefficients has dual advantages. The parameterised construction can be used to build a key scheme while the rational coefficients of the DWT enable an efficient hardware architecture using fixed point arithmetic (Pande and Zambreno, 2009). We get the following expression for $H_1(z)$ and $H_2(z)$.

$$\begin{aligned} H_1(z) = {} & \left(-9/64a + 1/32a^2 + 15/64 - 1/8/a\right) \\ & \left(z^4 + 1/z^4\right) \\ & + \left(-1/16a^2 + 11/32a - 11/16 + 1/2/a\right) \\ & \left(z^3 + 1/z^3\right) \\ & + \left(1/8 - 1/2/a\right)\left(z^2 + 1/z^2\right) \\ & + \left(-11/32a + 1/16a^2 + 15/16 - 1/2/a\right) \\ & (z + 1/z) \\ & + \left(9/32a - 1/16a^2 - 7/32 + 5/4/a\right) \end{aligned}$$

$$\begin{aligned} H_2(z) = {} & \left(1/32 - 1/32a\right)\left(z^3 + 1/z^3\right) \\ & + \left(1/8 - 1/16a\right)\left(z^2 + 1/z^2\right) \\ & + \left(7/32 + 1/32a\right)(z + 1/z) + (1/4 + 1/8a) \end{aligned}$$

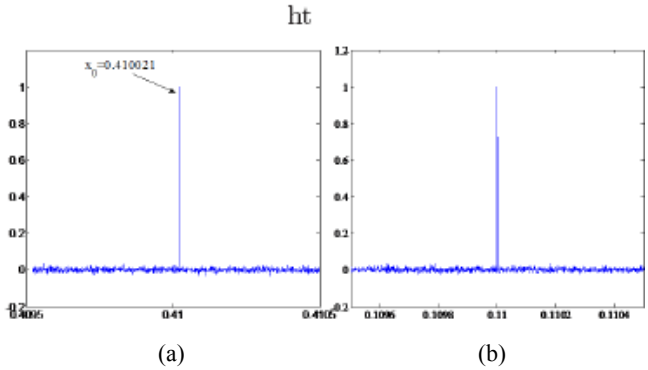
We get different DWT filters simply by changing the a values. The choice of the a value is secretly determined using a secret key. The numerical value of free parameter a can be varied over a wide range while retaining the perfect reconstruction property of the wavelet transform. However, as we vary the value of a over the range $(-\infty, +\infty)$, the output values of the DWT operation have a very large dynamic range requiring a larger number of bits for representation. This would reduce the compression rates achievable with the DWT-based coders. Numerical experiments show that parameterised DWT has a good PSNR value for image reconstruction with set-partitioning in Hierarchical Trees (SPIHT) based coder when a varies in the range 1 to 3. When a varies beyond this range, the output DWT coefficients are spread over a large dynamic range. At low bit rates, the encoder is not able to efficiently encode such a large range of input coefficients leading to poor compression results for natural images.

7 Security enhancement

A serious drawback of chaotic crypto-systems is that they are weak against known-plaintext attacks. If the plain-text and the ciphertext 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 has many advantages over logistic map:

- The MLM has better security properties than the logistic map. Figure 7 shows the sensitivity of MLM to the initial conditions. A slight difference in the initial condition leads to outputs which are completely uncorrelated. The bifurcation map for LM and MLM are shown in Figure 8. The absence of any white space in the key space of MLM allows us to build a continuous key-space. Figure 9 shows the graph for Lyapunov exponent for MLM which is higher than LM. A positive and higher Lyapunov exponent indicates the rate of divergence of two closely related inputs for the system.
- The random feedback scheme makes it difficult to predict the key value XORed to the original plaintext.
- The sequences s_n and p_n are linearly uncorrelated from each other making it difficult to reverse engineer the values of p_n from s_n.
- The sequence p_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 p_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.
- 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.
- Use of DWT parameterisation adds to the security of the scheme. The exact choice of DWT filter is given by a secret key. Lack of this knowledge will lead to inexact extraction of plain-text after decrypting the cipher-text.

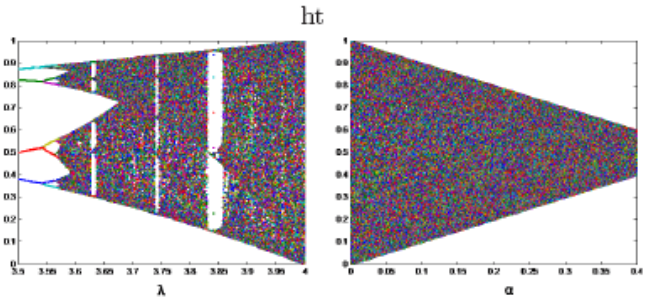
Figure 7 Correlation test of the pseudo-random sequence (a) generated using different initial values x_0 and (b) different initial parameter α



(a) (b)

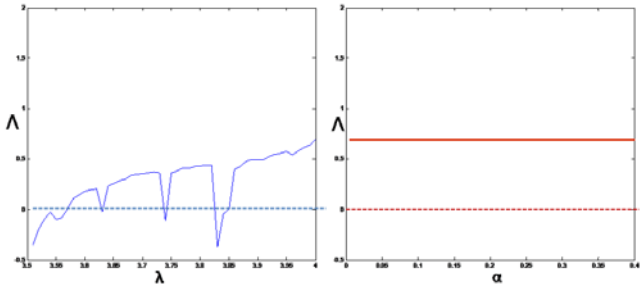
Note: The plots are measured against initial value $\alpha = 0.110000$ and $x_0 = 0.410021$.

Figure 8 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 (see online version for colours)



(a) (b)

Figure 9 Plot of Lyapunov coefficient (Λ – solid line) for (a) Logistic map as a function of parameter λ_{LM} indicating regions of non-chaotic behaviour and (b) modified logistic map showing higher divergence than logistic map and independence of Λ from parameter α (see online version for colours)



(a) (b)

The ICO shows good results against runs test, serial test, correlation test etc. which are used to prove the randomness of output $s[n]$ or s_n.

8 Hardware implementation

Figure 10 shows the hardware architecture for MCFB scheme. The input $x[n]$ is first pipelined for eight cycles and then the parameterised DWT filter is applied over it. The nine pipelined stages are then reduced to five by adding the stages with similar wavelet coefficients together to get $w_i[n]$ ($w_i[n] = x[n + i] + x[n - i]$, $i \in \{0, 4\}$). These are then multiplied with the a, a−1 and a2 values and summed up to get the low pass and high pass values $y_0[n]$ and $y_1[n]$. The outputs of two ICO s is then added to these two signals to get $z_0[n]$ and $z_1[n]$ respectively.

The hardware architecture of ICOs is shown in Figure 11. Two instances of ICOs are required in the design.

Some optimisation steps performed to reduce the cost of the underlying hardware are summarised below:

1 Division by binary coefficients (e.g., 1/64, 1/16, 1/4) was performed using arithmetic shift operations.

2 The input stream was pipelined. As shown in Figure 10, our architecture takes one pixel (or channel input) as the input and outputs the low and high pass signal coefficients with a finite latency. Increasing the system latency allows us to achieve a higher clock speed (and hence higher throughput).

Figure 10 Hardware architecture for the modified chaotic filter bank scheme (see online version for colours)

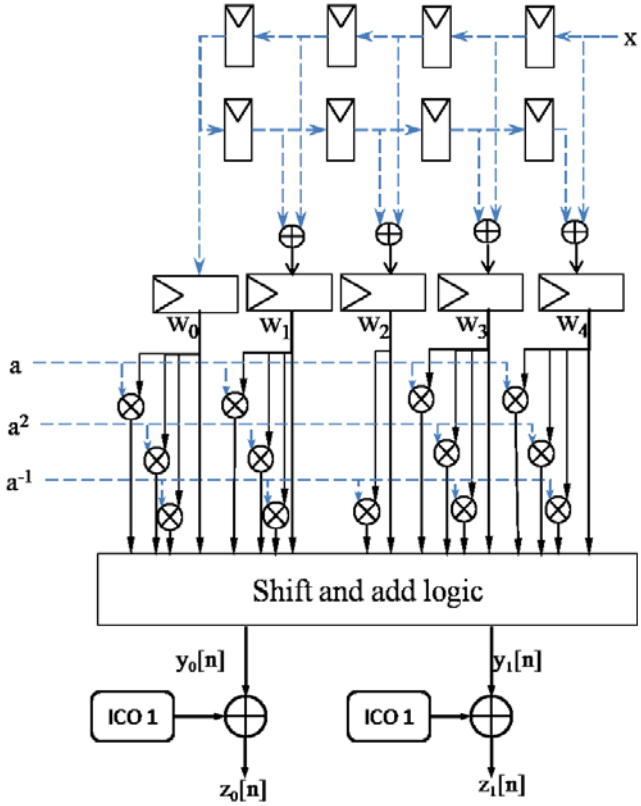
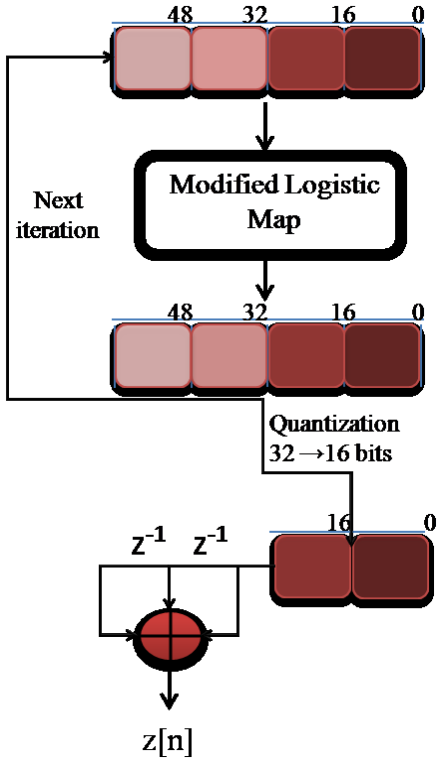


Figure 11 Hardware architecture for improved chaotic oscillator (see online version for colours)



The hardware implementation of proposed architecture was done using the Xilinx ISE 10.1 tool. The target device is a Xilinx Virtex-5 XC4VLX330 FPGA. The input $x[n]$ is 8 bits wide, the intermediate values $y_i[n]$ and $z_i[]$ are represented in 16 bits precision. The chaotic oscillator is implemented with an internal bit width of 64 bits, while only last 16 bits of the output of MLM contribute to the pseudorandom number generated by ICO. This prevents any cryptanalysis of ICO while requiring some extra computations. The 16 bit output of each ICO is added to the outputs $y_i[n]$ to get the output signal $z_i[n]$. Modulating the amplitude of ICO output ($s_i[n]$) allows us to change the range of the subband signal power to the chaotic subband power dynamically.

As mentioned, the iterating value of MLM ($x(i)$ and the parameters λ and μ are both implemented with 64 bits fixed-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 Integer.Floating point (I.F) format. The range for parameter λ is then calculated to be (Robilliard et al., 2006; Masuda et al., 2006) 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 $x(i)$.

A direct implementation gave a clock frequency of 67.8 MHz while requiring 48 DSP48E slices present in the Virtex-5 FPGA for efficient multiplication and addition operations. We present two optimisations to improve the clock frequency of the design while reducing the hardware requirements of the design.

8.1 Reconfigurable constant multiplier

The values a, a^2 and a^{-1} remain constant in the Parameterised DWT architecture for thousands of clock cycles. For example, in case of image processing, we will use the same a value for individual frame. Thus, the 13 multipliers used in the design can be replaced by reconfigurable look-up tables (LUTs) to allow fast arithmetic and more efficient implementation.

If the input is represented by B_1 bits and constant (a values) is represented by B_2 bits, we can use $(B_1 + B_2)$ B_2-input LUTs to get the output values of $H_1(k)$ and $H_2(k)$. Alternatively we can break down a $(B_1 \times B_2)$ bit multiplication into smaller input LUTs. Thus, the LUTs based multiplication can be reconfigured to incorporate any changes in encryption key (Pande and Zambreno, 2010).

Arbitrary hardware multipliers can be implemented using the propagate and generate algorithm (Mano and Ciletti, 2006). It is found that output is a function of inputs and is characterised uniquely by a logical expression which can be fit into a LUT. If one of the inputs (say B) is a constant, the output bit S_i can be represented as a logic function of bit values of the other input A.

$$S_i = f_i\left(A_n, A_{n-1}, \ldots A_1, A_0\right)$$

The truth table of these functions $f_i(...)$ can be evaluated either by logical simplification or by exhaustive search over the input values. We can implement a $M \times K$ bit constant multiplication using $(M + K)$ K-input LUTs. Next, we discuss the mapping of an $M \times K$ bit constant multiplier into 4-LUTs which are more freely available in commercial FPGAs.

8.1.1 Mapping a generic RCM into LUTs

The multiplication of two inputs A and B (M-bit variable input A, K-bit reconfigurable constant B) can be mapped to LUTs similar to 4 × 4 bits multiplier by obtaining a generic expression for S_1, S_2 ... S_{M+K-1}. S_i values can be represented as $f(A_{M-1}, A_{M-2}, \ldots A_1)$ and can be therefore mapped into an M-input LUT. We have $(M + K - 1)$ S_i values, requiring $(M + K - 1)$ M-input LUTs to multiply A and B.

A $(K + 1)$-input LUT can be built from 2 K-input LUTs. For example, we can build a 8-LUT from 2 7-LUTs which can be synthesised from 2 × 2 = 4 6-LUTs. Thus, one 8-LUT can be made from $2^4 = 16$ 4-LUTs and an arbitrary M-LUT from 2^{M-4} 4-LUTs.

Figure 12 gives an example of multiplication of 8-bit number with 12-bit constant ($M = 8$, $K = 12$). Figure 12(a) depicts implementation using 8-LUTs. 20 8-LUTs or equivalently 128 4-LUTs are used in the design.

Figure 12(b) provides an alternative implementation of the same multiplication by breaking the input number into multiples of 4-bit values. 4-input LUTs are used to obtain the X and Y values which are then added together using an adder. This implementation requires 32 4-LUTs and a 20 bit adder. This design requires less LUTs but the presence of 20-bit adder may slow down the clock speed of such a design.

Figure 12 Illustration of 12-bit constant multiplication with a 8-bit input (a) the individual bits of product are obtained as output of a 8-LUT (b) 4-LUTs are used in the implementation with the input A divided into two 4-bit values (see online version for colours)

```
          110110010001   (Operand 1)
        x     AAAAAAAA   (Operand 2)
SSSSSSSSSSSSSSSSSSSS   (Product  )
```

(a)

```
          110110010001   (Operand 1)
        x     AAAAaaaa   (Operand 2)
      XXXXXXXXXXXXXXXX   (aaaa * 0001)
+YYYYYYYYYYYYYYYY        (AAAA * 1001)
SSSSSSSSSSSSSSSSSSSS   (Product  )
```

(b)

8.2 Hardware optimisations for ICO

A single DSP48E slice can perform a maximum of 25 × 18 bits multiplication and hence 12 slices are required for a 64 × 64 bits multiplication. Two multiplications require 24 DSP48E slices.

We present an optimisation 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 $X_aX_bX_c$ where X_a,X_b and X_c are each 24 bit long sequences.

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

Similarly, we can represent Y as combination of four 16 bit numbers $Y_wY_xY_yY_z$.

$$\{Y\}_0^{71} = \{Y_w\}_{32}^{63}\{Y_x\}_{32}^{47}\{Y_y\}_{70}^{31}\{Y_z\}_0^{15}$$

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:

$$\begin{aligned} X \times Y = &\left(X_a \times 2^{48} + X_b \times 2^{24} + X_c\right) \times \left(Y_w \times 2^{48}\right. \\ &\left. + Y_x \times 2^{32} + Y_y \times 2^{16} + Y_z\right) \\ \Rightarrow X \times Y = &\, 2^{96} \times X_aY_w + 2^{72} \times X_bY_w + 2^{48} \times X_cY_w \\ &+ 2^{80} \times X_aY_x + 2^{56} \times X_bY_x + 2^{32} \times X_cY_x \\ &+ 2^{64} \times X_aY_y + 2^{40} \times X_bY_y + 2^{16} \times X_cY_y \\ &+ 2^{48} \times X_aY_z + 2^{24} \times X_bY_z + 2^0 \times X_cY_z \end{aligned}$$

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 DSP48E slice.

Thus,

$$\begin{aligned} X \times Y = &\, 2^{96} \times X_aY_w + 2^{76} \times X_bY_w + 2^{48} \times X_cY_w \\ &+ 2^{80} \times X_aY_x + 2^{56} \times X_bY_x \\ &+ 2^{64} \times X_aY_y + 2^{40} \times X_bY_y \\ &+ 2^{48} \times X_aY_z \end{aligned}$$

The other multiplication operation can also be optimised in a similar manner. Thus, we can reduce the hardware requirements and critical path for the implementation.

The above mentioned optimisations enhance the performance of original design. The use of reconfigurable LUTs instead of multipliers reduces the critical path of DWT architecture by replacing a multiplication operation with a look-up operation. The second optimisation – truncating the extra hardware for building ICO reduces the number of DSP slices used by the design by 33%.

The original design required 14 10 × 9 bits multipliers and four 64 × 64 bits multiplier which required 48 DSP48E slices and LUTs for implementation. The optimised implementation uses only 32 24 × 16 bits multiplier which are implemented in 32 DSP48E slices. Moreover, the achievable clock frequency increases by 30% from 67.8 MHz to 88.3 MHz.

9 Conclusions

This paper presents a novel chaotic filter bank based scheme for cryptographic operations. The scheme, based on MLM is suitable for embedded real-time applications and resistant to known cryptanalysis. The scheme can be used with image compression algorithms such as JPEG 2000.

This paper also presents a reconfigurable hardware implementation of the proposed scheme. Use of reconfigurable hardware allows partial removal of hard-multipliers from the design and gives improvement in clock frequency by 30%. The hard-coded key parameters (a values) can be changed by the use of partial reconfiguration techniques.

References

Arroyo, D., Li, C., Li, S. and Alvarez, G. (2009) 'Cryptanalysis of a computer cryptography scheme based on a filter bank', *Chaos, Solitons & Fractals*, Vol. 41, No. 1, pp.410–413.

Baptista, M.S. (1998) 'Cryptography with chaos', *Physics Letters*, Vol. 240, Nos. 1–2, pp.50–54.

Benkrid, A., Benkrid, K. and Crookes, D. (2003) 'Design and implementation of a generic 2D orthogonal discrete wavelet transform on FPGA', in *Proc. IEEE Symp. Field-Programmable Custom Computing Machines (FCCM)*, April, pp.162–172.

Benkrid, A., Crookes, D. and Benkrid, K. (2001) 'Design and implementation of a generic 2D biorthogonal discrete wavelet transform on an FPGA', in *Proc. IEEE Symp. Field-Programmable Custom Computing Machines (FCCM)*, pp.190–198.

Biham, E. (1991) 'Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91', in *Advances in Cryptology EUROCRYPT 91, Lecture Notes in Computer Science*, pp.532–534.

Carroll, T. and Pecora, L. (1991) 'Synchronizing chaotic circuits', *IEEE Transactions on Circuits and Systems*, April, Vol. 38, No. 4, pp.453–456.

Christopoulos, C., Skodras, A. and Ebrahimi, T. (2000) 'The JPEG2000 still image coding system: an overview', *IEEE Trans. Consumer Electronics*, November, Vol. 46, No. 4, pp.1103–1127.

Claus, C., Stechele, W., Kovatsch, M., Angermeier, J. and Teich, J. (2008) 'A comparison of embedded reconfigurable video-processing architectures', pp.587–590.

Gall, D.L. and Tabatabai, A. (1988) 'Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques', in *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.761–764.

Guanrong Chen, Y.M. and Chui, C.K. (2004) 'A symmetric image encryption scheme based on 3D chaotic cat maps', *Chaos, Solitions and Fractals*, vol. 21, no. 3, pp. 749–761.

Habutsu, T., Nishio, Y., Sasase, I. and Mori, S. (1991) 'A secret key cryptosystem by iterating a chaotic map', in *Advances in Cryptology EUROCRYPT 91, Lecture Notes in Computer Science*, pp.127–140.

Kocarev, L. (2001) 'Chaos-based cryptography: a brief overview', *IEEE Circuits and Systems Magazine*, Vol. 1, No. 3, pp.6–21.

Kocarev, L., Jakimoski, G., Stojanovski, T. and Parlitz, U. (1998) 'From chaotic maps to encryption schemes', in *IEEE Intl. Symp. Circuits and Systems*, June, Vol. 4, pp.514–517.

Koch, D., Beckhoff, C. and Teich, J. (2008) 'ReCoBus-Builder – a novel tool and technique to build statically and dynamically reconfigurable systems for FPGAs', in *Proc. IEEE Intl. Conf. Field Programmable Logic and Applications, FPL 2008*, Heidelberg, Germany.

Kotteri, K., Barua, S., Bell, A. and Carletta, J. (2005) 'A comparison of hardware implementations of the biorthogonal 9/7 DWT: convolution versus lifting', *IEEE Trans. Circuits and Systems II*, Vol. 52, No. 5, pp.256–260, May.

Liang, X., Zhang, J. and Xia, X. (2008) 'Improving the security of chaotic synchronization with a delta-modulated cryptographic technique', *IEEE Trans. Circuits and Systems II*, July, Vol. 55, No. 7, pp.680–684.

Ling, B.W-K., Ho, C.Y-F. and Tam, P.K-S. (2007) 'Chaotic filter bank for computer cryptography', *Chaos, Solitons & Fractals*, Vol. 34, No. 3, pp.817–824.

Mano, M.M. and Ciletti, M.D. (2006) *Digital Design*, 4th ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Martina, M. and Masera, G. (2007) 'Multiplierless, folded 9/7 – 5/3 wavelet VLSI architecture', *IEEE Trans. Circuits and Systems II*, September, Vol. 54, No. 9, pp.770–774.

Masuda, N. and Aihara, K. (2002) 'Cryptosystems with discretized chaotic maps', *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, January, Vol. 49, No. 1, pp.28–40.

Masuda, N., Jakimoski, G., Aihara, K. and Kocarev, L. (2006) 'Chaotic block ciphers: from theory to practical algorithms', *IEEE Trans. Circuits and Systems I*, June, Vol. 53, No. 6, pp.1341–1352.

Pande, A. and Zambreno, J. (2009) 'An efficient hardware architecture for multimedia encryption and authentication using discrete wavelet transform', in *IEEE CS Intl. Symp. VLSI*, pp.85–90.

Pande, A. and Zambreno, J. (2010) 'A reconfigurable architecture for secure multimedia delivery', in *IEEE 23rd International Conference on VLSI Design*.

Pichler, F. and 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, Springer-Verlag, London, UK.

Ritter, J. and Molitor, P. (2001) 'A pipelined architecture for partitioned DWT based lossy image compression using FPGAs', in *Proc. Intl. Symposium on Field Programmable Gate Arrays (FPGA)*, pp.201–206.

Robilliard, C., Huntington, E. and Webb, J. (2006) 'Enhancing the security of delayed differential chaotic systems with programmable feedback', *IEEE Trans. Circuits and Systems II*, August, Vol. 53, No. 8, pp.722–726.

Rueppel, R. (1986) *Analysis and Design of Stream Ciphers*, Springer, Berlin.

Shannon, C.E. (1949) 'Communication theory of secrecy systems', *Bell Systems Technical Journal*, Vol. 28, pp.656–715.

Tseng, P., Chang, Y., Huang, Y., Fang, H., Huang, C. and Chen, L. (2005) 'Advances in hardware architectures for image and video coding – a survey', *Proc. IEEE*, January, Vol. 93, No. 1, pp.184–197.

Vetterli, M. and Kovačevic, J. (1995) *Wavelets and Subband Coding*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Yang, T. (2004)' A survey of chaotic secure communication systems', *International Journal of Computational Cognition*, Vol. 2, No. 2.

Yaobin Mao, G.C. and Lian, S. (2004) 'A symmetric image encryption scheme based on 3D chaotic baker maps', *Intl. J. Bifurcat Chaos*, Vol. 14, No. 10, pp.3613–3624.