J Real-Time Image Proc
DOI 10.1007/s11554-010-0165-6

ORIGINAL RESEARCH PAPER

The secure wavelet transform

Amit Pande · Joseph Zambreno

Received: 21 August 2009 / Accepted: 28 May 2010

Abstract There has been an increasing concern for the security of multimedia transactions over real-time embedded systems. Partial and selective encryption schemes have been proposed in the research literature, but these schemes significantly increase the computation cost leading to tradeoffs in system latency, throughput, hardware requirements and power usage. In this paper, we propose a lightweight multimedia encryption strategy based on a modified discrete wavelet transform (DWT) which we refer to as the secure wavelet transform (SWT). The SWT provides joint multimedia encryption and compression by two modifications over the traditional DWT implementations: (a) parameterized construction of the DWT and (b) subband re-orientation for the wavelet decomposition. The SWT has rational coefficients which allow us to build a high throughput hardware implementation on fixed point arithmetic. We obtain a zero-overhead implementation on custom hardware. Furthermore, a Look-up table based reconfigurable implementation allows us to allocate the encryption key to the hardware at run-time. Direct implementation on Xilinx Virtex FPGA gave a clock frequency of 60 MHz while a reconfigurable multiplier based design gave a improved clock frequency of 114 MHz. The pipelined implementation of the SWT achieved a clock frequency of 240 MHz on a Xilinx Virtex-4 FPGA and met the timing constraint of 500 MHz on a standard cell realization using 45 nm CMOS technology.

A. Pande (✉) · J. Zambreno
Electrical and Computer Engineering, 2215 Coover Hall,
Ames, IA 50011, USA
e-mail: amit@iastate.edu

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

Keywords Discrete wavelet transform · Multimedia encryption · Reconfigurable hardware

1 Introduction

The recent emergence of embedded multimedia applications, such as mobile-TV, surveillance, video messaging, and tele-medicine have increased the scope of multimedia in our personal lives. These applications increase the concerns regarding privacy and security of the targeted subjects. Another growing concern is the protection and enforcement of intellectual property rights for images and videos. These and other issues such as image authentication, rights validation, identification of illegal copies of a (possibly forged) image are grouped and studied under the label of digital rights management (DRM).

The computer security protocols (e.g. SSL, Wagner and Schneier [34], TLS [4]) and cryptographic ciphers, e.g. AES (FIPS 197 [9]), DES (FIPS 46-2 [10]), IDEA [16], drive much of the world's electronic communications, commerce, and storage. These techniques have been used for conventional multimedia encryption and authentication.

In one version of these schemes, some form of private-key encryption algorithm is applied over the full or partial output bit stream from the video compression engine. This naive approach is usually suitable for text, and sometimes for small bitrate audio, image, and video files that are being sent over a fast dedicated channel. Secure Real-time Transport Protocol, or shortly SRTP [2]), is an application of the naive approach. In SRTP, multimedia data is packetized and each packet is individually encrypted using AES. The naive approach enables the same level of security as that of the used conventional cryptographic cipher.

Published online: 06 July 2010

Consequently, a multimedia compression engine [32] has an additional encryption engine to ensure multimedia security. Depending on the scheme used, the encryption operation is performed either at some intermediate level during compression or after the final compression. However, these cryptographic ciphers require a large amount of computational resources and often incur large latencies. Hardware implementations of AES are often pipelined, leading to a significantly large latency for real-time applications [12].

The large data volumes, interactive operations, real-time responses, and scalability features that are inherent to real-time multimedia delivery restrict the practical application of these naive cryptographic schemes. Selective encryption schemes have been proposed in research literature [19]; Lian et al. [18]; Martin and Plataniotis [24]; Brachtl et al. [3]; Engel and Uhl [8] to reduce the computational requirements of full encryption schemes. Lian et al. [18] present a scheme for encryption of Discrete Cosine Transform (DCT) coefficients' signs and watermarking of DCT coefficients. Lian et al. [18] uses Exp-Goloumb codes for the encryption operation. Cheng and Li [6] propose a DWT-based partial encryption scheme which encrypts only a part of compressed data. Only 13–27% of the output from quadtree compression algorithms is encrypted for typical images. A good summary of efforts in selective or partial encryption of images can be found in Liu and Eskicioglu [19].

Furthermore, embedded multimedia systems have constraints on power consumption, available computation power and performance. Real-time embedded systems face additional constraints on power consumption, hardware size and heat generation in the chip which requires design and mapping of computation-savvy encryption schemes for such architectures. Recently, power-aware designs have been proposed for video coding in embedded scenarios [5]. The authors in Cheng et al. [5] propose a multi-mode embedded video codec with DRAM area and external access power savings to support a real-time encoding of CIF images (having resolution of 352×288 pixels). Adding a sequential or pipelined encryption stage to the system in Cheng et al. [5] will add to system latency and further increase the power/heat budget of such a design.

Such limitations can be alleviated through the development of parameterized compression blocks that can achieve simultaneous encryption. Thus, the compression operation itself uses a key to encode the input data and no external cryptographic engine is required. Recently, some schemes have been developed using this compression-combined-encryption approach. Grangetto et al. [11] introduce a parameterization in the arithmetic coding stage of multimedia compression. This parameterization is used to build a key scheme. However, the performance of such scheme for embedded systems remains untested. Kim et al. [15] presents a variation of [11] that improves the security performance of parameterized arithmetic coding scheme but increases the complexity in hardware implementation.

Mao and Wu [22] presents a joint signal processing and cryptographic approach to multimedia encryption. They use index mapping and constrained shuffling to achieve confidentiality protection. This ensures that the encrypted bitstream still complies with the state-of-the-art multimedia coding techniques. The scheme gives good results, however, it requires extra computations (and hence extra hardware resources) to implement such a scheme. Lian and Wang [17] presents a multimedia encryption scheme based on wavelet coefficients confusion. However, a scheme based on wavelet coefficients permutations alone is bound to be separable and weak against any cryptanalysis. In this work, we do use a wavelet coefficient permutation called 'subband re-orientation' which is optimized for implementation without any computation overheads. However, our overall scheme has more parameters that build the key space which prevents an adversary from easily cracking our scheme by parallel brute force trials in the individual subbands.

Fast Encryption Algorithm for Multimedia (FEA-M) has been proposed for real-time multimedia encryption [36]. It works with Boolean matrix and can be implemented efficiently on hardware. However, there have been several attacks against such algorithms and proposals have been written to improve the security [35].

This paper presents a multimedia encryption scheme based on parameterized construction of the DWT and subband re-orientation for the wavelet decomposition, called the secure wavelet transform (SWT). An efficient hardware implementation (direct implementation and a reconfigurable constant multiplier (RCM) based implementation) of the SWT using both FPGA and ASIC technology is also presented in this paper. The initial results regarding parameterized construction of the DWT were presented in Pande and Zambreno [26].

Section 2 gives the theory and mathematical preliminaries of the proposed SWT architecture. Section 3 discusses the image security provided by the SWT. In Sect. 4, we present an optimized hardware architecture for the SWT. Hardware optimizations, FPGA and ASIC implementation results and a RCM implementation has been presented in this section. Section 5 concludes the paper with insight of future works.

2 Preliminaries

Prior works in signal processing establish that the 1D DWT can be viewed as a signal decomposition using specific low pass and high pass filters Strang and Nguyen [31]. 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, respectively. After each filtering stage, down sampling is done by a factor of two to remove the redundant information.

The two most common DWT filters used in image compression are the Le Gall's 5/3 and the Daubechies 9/7 filters Christopoulos et al. [7], accepted in the JPEG2000 standard. The Le Gall's filter has rational coefficients and its hardware implementation requires less resources. The Daubechies 9/7 filter has better compression performance, however, it has irrational coefficients and leads to lossy compression. Applying a 2D DWT to an image of resolution $M \times N$ results in four images of dimensions $\frac{M}{2} \times \frac{N}{2}$. Subsequent levels of DWT-based decomposition yield a multi-resolution structure suitable for image compression.

2.1 Parameterized construction of DWT

There are four filters that comprise the two-channel bi-orthogonal wavelet system. The analysis and synthesis low-pass filters are denoted by H_1 and H_2, respectively. The analysis and synthesis high pass filters are denoted by G_1 and G_2, respectively, and are obtained by quadrature mirroring the low-pass filters.

$$G_1(z) = z^{-1}H_2(-z), G_2(z) = zH_1(-z)$$

The perfect reconstruction (PR) condition for a DWT filter simplifies to the following:

$$H_1(z)H_2(z) + H_1(-z)H_2(-z) = 2$$

Liu and Zheng [20] present a parameterized construction of Bi-orthogonal wavelets filter banks (typically used for image compression). For even number of vanishing moments, $H_1(z)$ and $H_2(z)$ are represented as follows:

$$H_1(z) = \left(z^{-\frac{1}{2}} + z^{\frac{1}{2}}\right)^{2l_1} \times \left(\alpha + (1-\alpha)\left(z^{\frac{1}{2}} + z^{\frac{1}{2}}\right)^2\right)$$

$$H_2(z) = \left(z^{-\frac{1}{2}} + z^{\frac{1}{2}}\right)^{2l_2} \times Q(z)$$

where

$$Q(z) = \sum_{n=0}^{3} q_n \times \left(z^{\frac{1}{2}} + z^{\frac{1}{2}}\right)^{2n}, l_1, l_2 \geq 0, \{l_1, l_2\} \in Z$$

and α is the free parameter introduced in the design. The values q_n are calculated by the following expression:

$$q_n = \sum_{k=0}^{n} \left(\binom{L+n-k-1}{L-1} [2(1-\alpha)]^k \right),$$
$$n = 0, \ldots, L-1$$

and

$$q_L = \frac{1}{2\alpha}\left\{ \binom{2L-k-1}{L-1} [2(1-\alpha)]^k + (1-2\alpha) \sum_{n=0}^{L-1} q_n \right\}$$

with $L = l_1 + l_2$.

For the 9/7 filter, the values of q_n were approximated using Taylor's series expansion and obtained as follows:

$$q_0 = 1;\ q_1 = 5 - 2\alpha;\ q_2 = 4\alpha^2 - 14\alpha + 16;$$

$$q_3 = 36\alpha - 8\alpha^2 - 60 + 32/\alpha;$$

Simplifying these equations, we get the following expression for $H_1(z)$ and $H_2(z)$.

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

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

There are several useful features of parameterized DWT construction that make it suitable for being a part of the SWT:

2.1.1 Rational coefficients

The expressions for $H_1(z)$ and $H_2(z)$ have product of exponents in α and z with rational coefficients. All these rational coefficient multiplication operations can be simplified into shift-add operations. For example, $\frac{A}{16} \equiv A \triangleright 4$ and $\frac{15B}{64} \equiv (B \triangleright 2) - (B \triangleright 6)$ where $\triangleright$ denotes a right shift operation.

2.1.2 Feasible range of parameter α

The numerical value of free parameter α can be varied over a wide range while retaining the perfect reconstruction property of the wavelet transform. However, as we vary the value of α 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 parameterized DWT has a good PSNR value for image reconstruction with set-partitioning in hierarchical trees (SPIHT) based coder when α varies in the range 1–3. When α 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. Figure 1 illustrates the significant decline in PSNR values (in db) for $\alpha > 3$.

2.1.3 Key-space

We divide this interval [1, 3] into 2^m sub-intervals. Thus, a 1D DWT operation is represented by m bits. One level of wavelet decomposition involves successive filtering with row and column filters. If we have N levels of decomposition using DWT, we can choose different α values for all $2N$ filters (represented by 16 mN bits).

The actual choice of N and the number of sub-intervals is subjective and depends on input images and desired sensitivity of images. For example, the image sequences which are input to highly crucial image processing applications, such as medical imaging, can use more sub-intervals while some applications—such as counting the number of cars crossing an intersection—will allow low number of bits. Figure 2 shows the mean square values (MSE) for image encoded with one α value and reconstructed with the adjacent α value for various bit-width. It can be seen that 5 or less bits give a large MSE (MSE > 8) while some applications may allow $m = 8$.

Figure 3 shows the image performance of the parameterized DWT. We took three sample images: the first and third being an aerial survey of some landscape while the second image is a snapshot of Shakespeare's written text (Scene II from Julius Caesar). The results are presented when an encryption (or image compression) was performed with the α parameter set to 2.0 and decryption (or image reconstruction) was performed with different α values. We can see that the images decrypted with the wrong key values (Fig. 3b, d, e) have poor visual quality. These images miss many important details of the original scene or

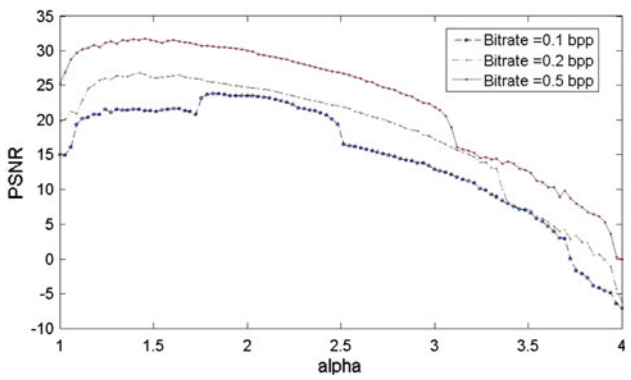

Fig. 1 PSNR values (in db) for image reconstruction using SPIHT coder at different bitrates (in bpp or bits per pixel)

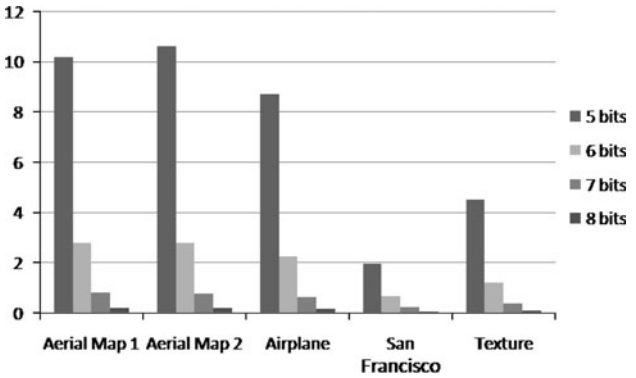

Fig. 2 MSE values for sample images with the change in number of bits assigned to one α parameter. The image was encoded with one α value and decoded with adjacent alpha values for various bit-widths of α. 1,000 simulations were run to obtain an average value

text. In this experiment, we have visualized the impact of only using the parameterized DWT and a single key for all levels of decomposition.

It can be seem that wrong guesses for DWT parameter α leads to high reconstruction errors in images. However, we need further dimensions to increase the key space and make image reconstruction more obscure in case of wrong guesses for the key value.

2.2 Subband re-orientation

The parent-child coding gain in the DWT-based coders was quantified by Marcellin and Bilgin [23]. These dependencies are generally credited for the excellent mean square error (MSE) performances of zero-tree-based compression algorithms such as embedded zero-tree coding of wavelet coefficients (EZW) and SPIHT. The subbands were rotated by 90° with respect to the previous scale prior to zero-tree coding. These experiments indicate that the coding gain due to these dependencies is not considerable for natural images (typically around 0.40 dB for SPIHT-NC and 0.25 dB for SPIHT-AC). However, the image reconstruction quality will considerably change with the rotations of subbands. Simple transformations, such as transposing the subband matrix, reverse-ordering of the subbands along the rows and columns, can be implemented in the subband images simply by modifying the memory access pattern of the computing block, without any computational overhead. Such simple modifications in subband orientation can highly affect image perception and can be implemented without any computational overheads. It can be used as a parameter for the SWT operation. A prior knowledge of these parameters is a must in order to decompress the image correctly. There are several useful features of subband re-orientation that make it suitable for being a part of the SWT:

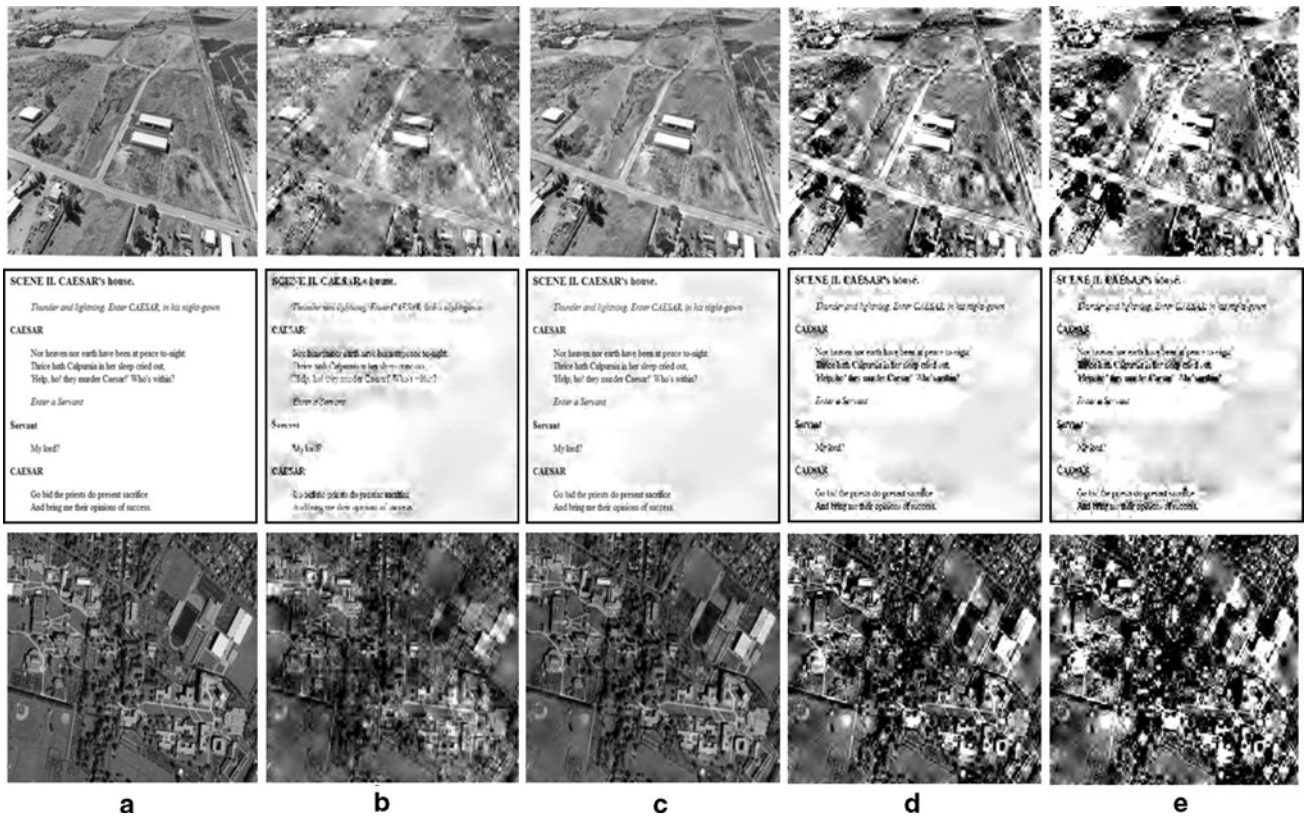

Fig. 3 Image reconstruction with different keys. **a** The original images which are then encrypted with $\alpha = 2$, **b–e** show reconstruction with $\alpha = 1$, 2, 3 and 3.5, respectively

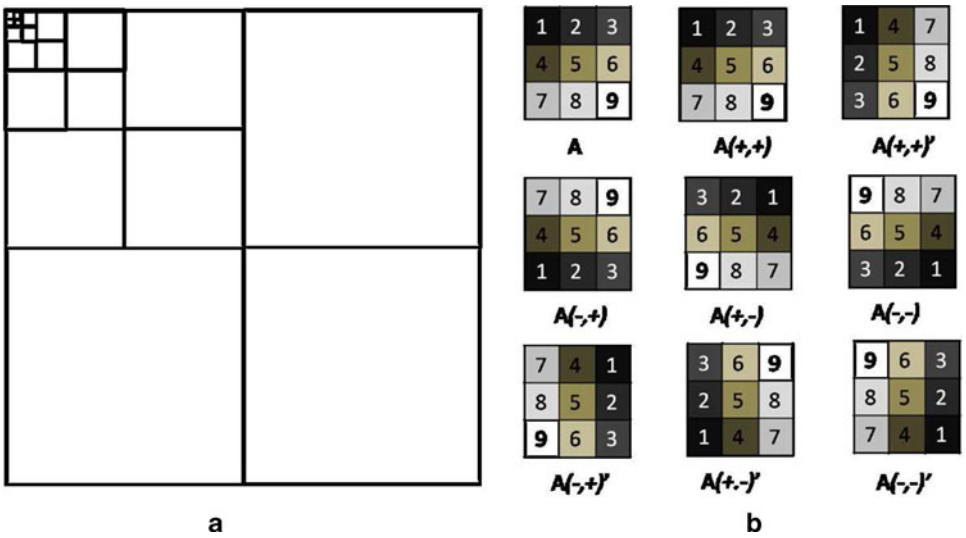

Fig. 4 **a** Image decomposition with DWT (6 levels) leading to 19 subbands. 3 bits are assigned for each subband's re-orientation information. **b** Possible transpose relationships for subbands. *A* is the original matrix. The eight permutations are achieved using transpose relationship('), and reverse-ordering of the subbands (− for reverse, + for forward read access) along both *rows* and *columns*

2.2.1 Zero computational overhead

Subband re-orientation can be achieved by intelligently writing the outputs of DWT filter to the memory without any overheads in computational costs of the system.

2.2.2 Feasible subband re-orientations

In Fig. 4b, we illustrate how we can represent the same subband in eight different orientations: we have four orientations of the subband decided by the forward or reverse

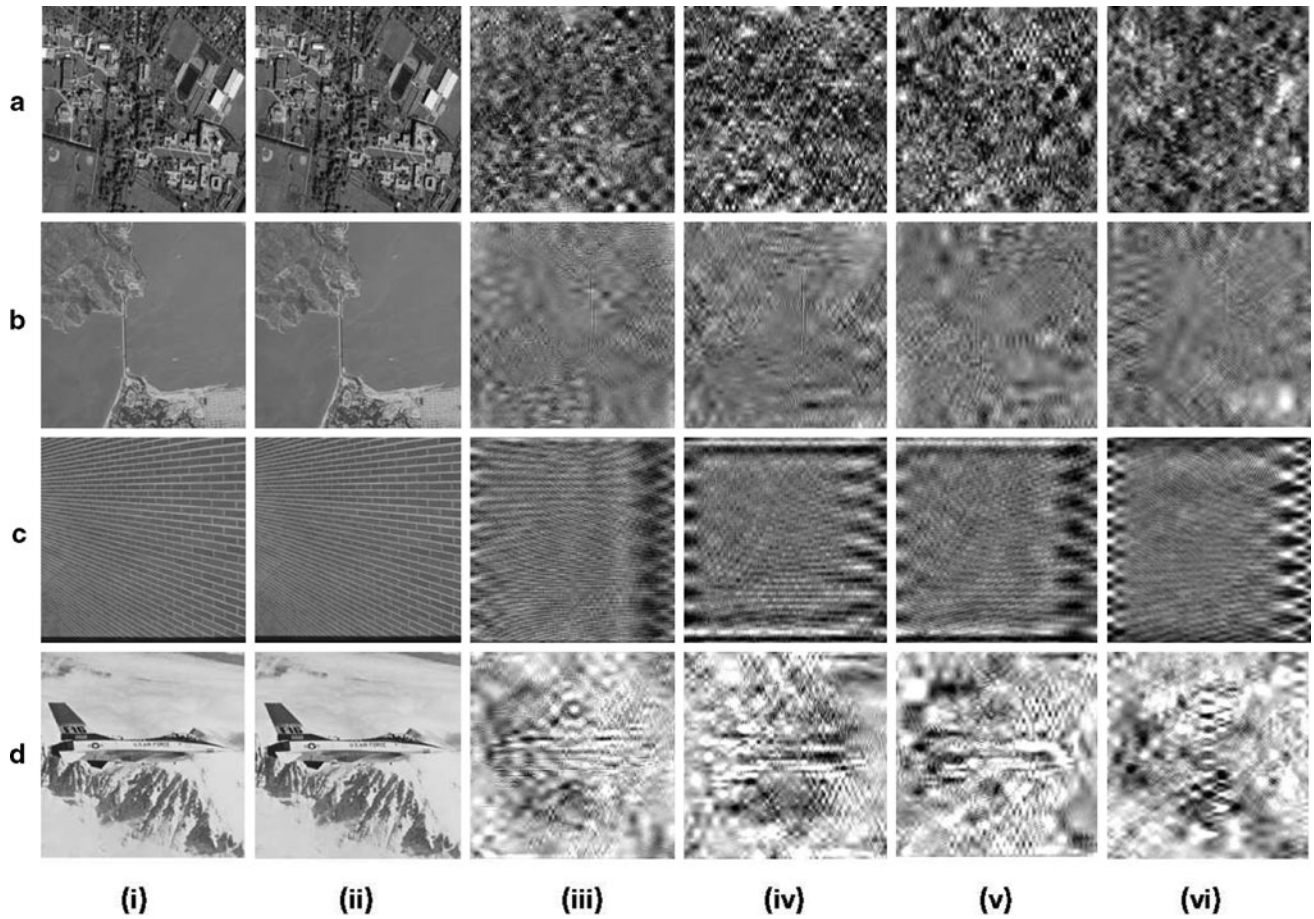

Fig. 5 Image reconstruction with different keys. *A* aerial map image, *B* San Francisco Golden gate aerial image, *C* brick wall (texture) image and *D* airplane image. *i* Original image encrypted with key0, *ii* Image decrypted with same key, *iii–vi* Image decrypted with randomly generated keys

ordering of the matrix along rows or columns. We get four more orientations by transposing the above four, summing up to eight possible transformations for each subband. We need a 3 bit value to represent this transformation for a single subband.

2.2.3 Key-space

Figure 4a shows the 19 different subbands obtained by a 6-level wavelet decomposition. In general, we obtain $3N + 1$ subbands for a N level wavelet decomposition, each requiring 3 bits. Thus, we get a keyspace of $9N + 3$ bits using subband re-orientation.

3 Security

In this section a brief evaluation of the security features of proposed scheme is presented. A key-space of 16 mN + $(9N + 3)$ bits can be obtained from N levels of wavelet decomposition. For an image size of 512 × 512 pixels this upper limit of N ($N_{\max}$) is 9. However, choosing N close to $N_{\max}$ will lead to the innermost subband size being very small.

We selected wavelet decomposition level of $N = 6$ for images of dimension 512 × 512 pixels to allow a standard block size of 8 × 8 pixels for the innermost subbands. $m = 8$ is set for applications sensitive to image quality while $m = 5$ works for all general applications.

Shannon's [29] paper , which serves as the foundational treatment of modern cryptography calls this property as the 'confusion' property. Ideally, change in one bit of the key should change the cipher text completely.

Figure 5 gives the performance of our scheme against attacks with random keys. The images decrypted with wrong keys have little resemblance to original images as indicated by the PSNR values for these reconstructed images (as shown in Table 1). Figure 6a–d gives the plot of PSNR values of reconstructed images for the four test images. 1,000 such trials were run with different random

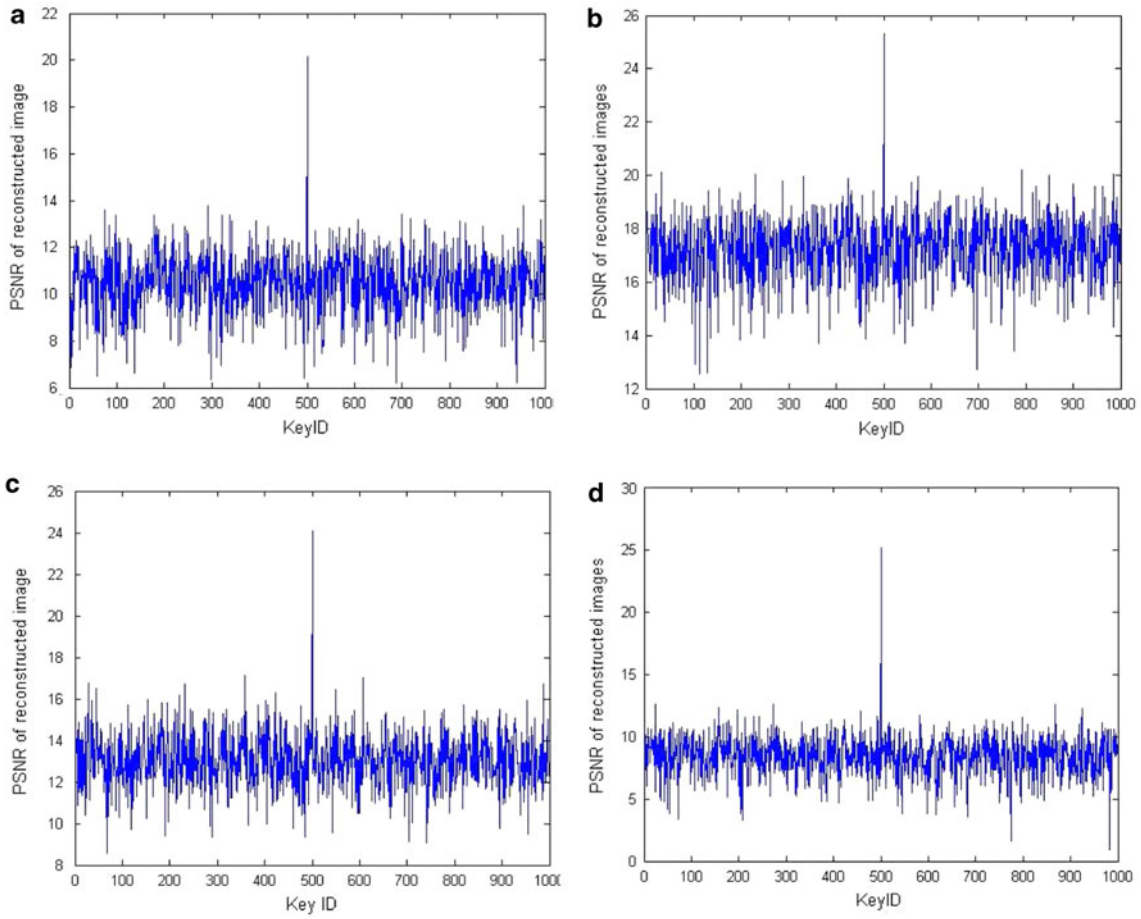

Fig. 6 Image reconstruction with randomly generated keys. **a**–**d** give result of 1,000 random trials on the four sample images respectively. The *x*-axis gives results with different keys. The 500th trial (with 500th key) refers to the test case with decryption with same key as the encryption key. The *y*-axis represents the PSNR value of the reconstructed images

Table 1 PSNR values (in db) for image reconstruction with various random keys (encoded with key0)

	Key0	Key1	Key2	Key3	Key4
Aerial	∞	12.36	11.17	11.67	11.77
San Francisco	∞	18.40	17.34	18.21	18.46
Brick Wall	∞	14.75	13.39	14.34	13.58
Airplane	∞	13.19	11.26	11.63	12.43

keys. The single peak in each graph is observed for the 500th trial where the original key (for encryption) and the decryption key are the same.

The hamming distance (*h.d.*) between two strings of equal length is the number of positions for which the corresponding symbols are different. i.e. the minimum number of bits that must be "flipped" to go from one word to the other. An ideal encryption scheme must give entirely random output if the *h.d.* between the encryption and decryption keys is non-zero. That is the case with block ciphers such as AES or DES which allow enough mixing between bit values in multiple rounds to achieve that effect. The performance of SWT, is thus going to be less than the conventional cryptographic schemes.

We tested our scheme for image reconstruction performance with small *h.d.* between the two keys. Our scheme provides security as evident by the low PSNR values (for $h.d. \geq 4$) in Table 2. 1,000 simulations were run to obtain the average PSNR value of reconstructed image with different hamming distances between the encoder and decoder key. It can be observed from the PSNR values that a hamming distance of 6 and above gives a perceptible reduction in image appearance (indicated by low PSNR value). The visual results are shown in Fig. 7. Different bit positions in the key have different effect on the image quality degradation. This is attributed to the fact that changing different bit positions in value of α will lead to different degrees of distortions. This attributes to the fact that Figure 7D(vi) has less quality degradation compared to Fig. 7D(v). To quantify the image degradation with increasing *h.d.*, we ran 1,000 simulations and recorded the average values in Table 2

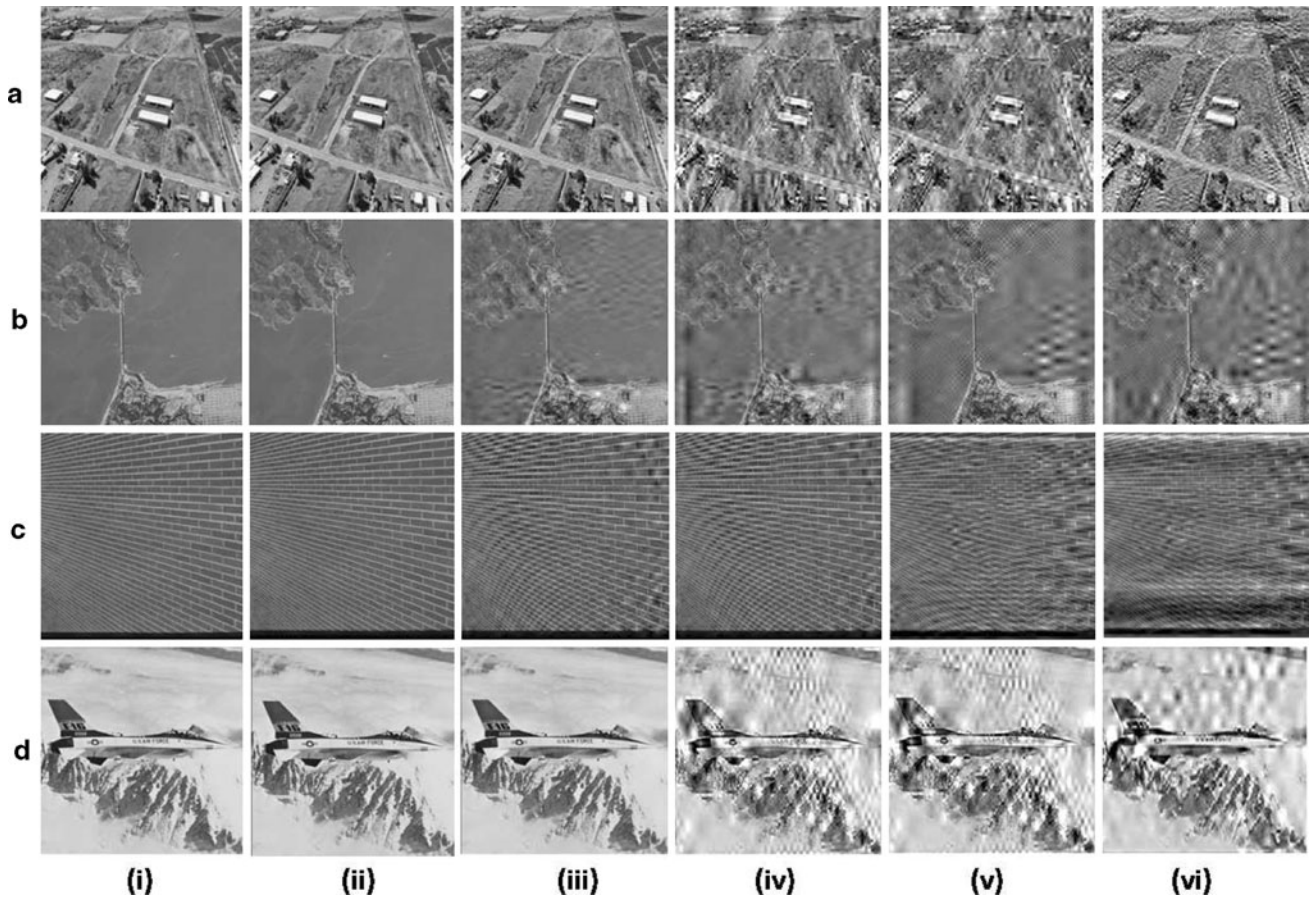

Fig. 7 Image reconstruction with different keys. *A* aerial map image, *B* San Francisco Golden gate aerial image, *C* brick wall (texture) image and *D* airplane image. *i* Original image encrypted with key0, *ii* Image decrypted with same key, *iii–vi* Image decrypted with hamming distance of 1,4, 6 and 8

Table 2 Variations in image reconstruction quality (PSNR values) with hamming distance

Hamming distance	0	1	4	6	8
Aerial	∞	50.3	23	16.04	13.18
San Francisco	∞	36.27	30.98	22.61	21.09
Brick Wall	∞	50.27	37.5895	25.9	23.2
Airplane	∞	44.28	21.64	21.43	16.16

4 Hardware implementation

Figure 8 gives an overview of the 1D SWT hardware architecture. The input data (one pixel input per cycle) x is pipelined for eight cycles. We observe that z^i and z^{-i} values in expressions for $H_1(z)$ and $H_2(z)$ have the same coefficients. Thus, these values can be added together to simplify further computations. In Fig. 8, eight of the nine inputs are passed through four adders to reduce the number of input to five. These values (labeled w_0, w_1, w_2, w_3 and w_4) are

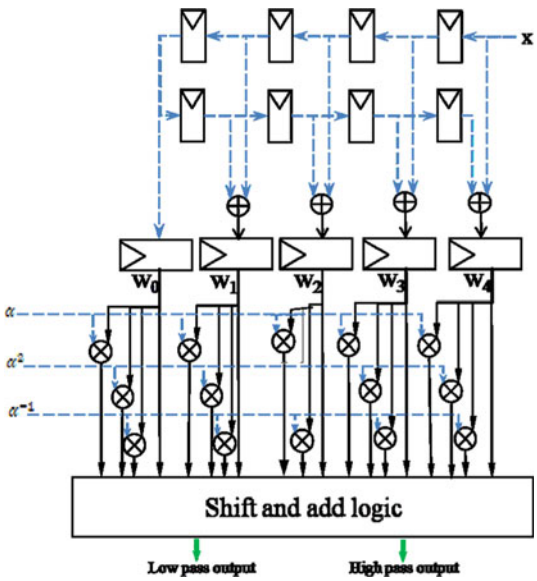

Fig. 8 Hardware Architecture for THE 1D SWT Filter

Table 3 Hardware utilization of DWT architecture on Xilinx Virtex XCVLX330 FPGA

	SWT (RCM)	SWT (VLSI)	SWT (FPGA)	Pande and Zambreno [27, 28]	Martina and Masera [25]	Daub-echies 9/7	Jou et al. [14]	Vishwanath et al. [33]	Huang et al. [13]
Slices flip flops	5580	–	649	245	–	210	–	–	–
Multipliers	0	14	14	0	0	16	12	36	12
Adders	11	41	41	9	19	15	16	36	16
Registers	120	–	92	208	–	144	–	–	–
Critical path	$4T_a + T_l$	$T_m + 5T_a$	$T_m + 5T_a$	$3T_a$	$5T_a$	$T_m + 4T_a$	$T_m + 2T_a$	$T_m + 4T_a$	$4T_m + 8T_a$
Clock frequency (MHz)	114 (np) –	– 500 (p)	60 (np) 243 (p)	120 (np)	200	107	–	–	–
Security	Yes	Yes	Yes	No	No	No	No	No	No

T_m and T_a are the time delay in multiplier and adder circuits, respectively

np not pipelined, *p* pipelined

multiplied with α, α^2 and α^{-1} to obtain the necessary intermediate values which are input to shift and add logic. The high and low pass filter coefficients are the final output of the 1D SWT filter.

We performed several optimization steps to reduce the cost of the underlying hardware. Division by binary coefficients (e.g. 1/64, 1/16, 1/4) was performed using arithmetic shift operations. This reduces the number of multipliers in the circuit from 69 to 23. Reducing the number of inputs from nine to five reduces the number of adders in the design from 70 to 41 and the number of multipliers from 23 to 14.

The initial work was presented in Pande and Zambreno [26, 27] and there is a mistake in the count of multipliers reported in them.

The input stream was then pipelined to achieve a higher clock frequency (and hence higher throughput).

A Xilinx XC5VLX330 FPGA was targeted for our experiments, using ModelSim 6.4 and Xilinx ISE 10.1 for simulations and synthesis. The non-pipelined design had clock frequency of 60 MHz while a pipelined design with four extra cycles of latency achieves a clock frequency of 242 MHz. The design was also implemented using Synopsys Design Compiler with the free PDK [30] 45 nm cell library. Under the timing constraints of 500 MHz, the design required 4,259 cells and a chip area of 22,066 μm^2.

The design used 14 10 $\times$ 9 bit multipliers, 41 adders (20 18-bit adders and 21 9-bit adders). The hardware requirements of our implementation are summarized and compared with other implementations in Table 3. The critical path of the implementation is $T_m + 5T_a$ where T_m indicates the time delay in multiplier and T_a indicates the time delay in adder circuit.

The subband re-orientation part in DWT is done by changing the write pattern of the subbands after the SWT operation. Thus, no computational overhead is involved in such an operation. It is noteworthy that ours is the first proposal for image and video security based on SWT and its hardware implementation.

The initial parameterized DWT design obtained a clock frequency of about around 60 MHz, due to its long critical path. The critical path of the circuit lies from the w_i registers to the final low pass output. We then pipelined this computation into several stages and obtained a faster implementation. By adding 4 pipelining stages we obtained a clock frequency of 242 MHz.

4.1 Reconfigurable constant multiplier (RCM)

The expression for low and high pass filter coefficients were obtained in Sect. 2.1. It was implemented in Fig. 8 using several multiplier units. The w_i, $i \in \{0, 1, 2, 3, 4\}$ values are obtained by summing the inputs for symmetric taps in the SWT implementation as shown in Fig. 8. w_i is calculated as follows:

$$w_i(k) = x(k+i) + x(k-i), \quad i \in \{0, 1, 2, 3, 4\}$$

Then, we can represent the filter expressions as:

$$H_1(k) = \sum_{i=0}^{4} K_i(a) \times w_i(k)$$

and

$$H_0(k) = \sum_{i=0}^{3} \hat{K}_i(a) \times w_i(k)$$

Here, $K_i(a)$ and $\hat{K}_i(a)$ are the functions of the variable a, and w_i are obtained from the pipelined input. The values of functions $K_i(a)$ and $\hat{K}_i(a)$ remains the same as long as we have the same a parameter. This implies that this value of these functions behave as a constant and changes only when we change the encryption key (and the associated parameter a). This value can thus be computed and hard-coded into the circuit. This constant multiplication can

easily be mapped to a reconfigurable hardware with programmable LUTs. If the input is represented by B_1 bits and constant is represented by B_2 bits. We can use $(B_1 + B_2)$ B_2-input LUTs to get the output values $H_1(k)$ 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 multiplier can be reconfigured corresponding to incorporate any changes in encryption key.

This idea is used to build a Reconfigurable Constant Multiplier or RCM. A RCM has one operand which is kept constant and mapped to LUTs while the other multiplicand is a variable and changes its value every clock cycle. The constant operand can be changed dynamically by reconfiguring the LUT values on-the-fly.

We discuss the implementation of a 4 × 4 bit RCM to explain the LUT mappings.

4.1.1 4 × 4 Bits multiplier using LUTs

Let A and B be the two operands, both being 4 bits long. Let us define two new binary variables:

$$P_i = A_i \oplus B_i, G_i = A_i B_i$$

The output bit and the sum at each stage can be represented as:

$$S_i = P_i \oplus C_i C_{i+1} = G_i + P_i C_i$$

On simplification Mano and Ciletti [21], we get

$$C_1 = \text{initial carry}$$
$$C_2 = G_1 + P_1 C_1 = A_1 B_1 + (A_1 \oplus B_1) A_0 B_0$$
$$C_3 = G_2 + P_2 G_1 + P_2 P_1 C_1$$
$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_1$$

and

$$S_1 = A_1 \oplus B_1 \oplus C_1 = A_1 \oplus B_1 \oplus A_0 B_0$$
$$S_2 = A_2 \oplus B_2 \oplus C_2 = A_2 \oplus B_2 \oplus (A_1 B_1 + (A_1 \oplus B_1) A_0 B_0) \ldots$$

We make some interesting observations and inferences.

- C_i values can be expanded and expressed in terms of A_i and B_i values.
- Similarly, a complex logical expression can be generated for each S_i value. Each S_i value is characterized uniquely by a logical expression.
- If one of the inputs (say B) is a constant, S_i can be represented as a logic function of bit values of A.

  $$S_i = f_i(A_3, A_2, A_1, A_0)$$

- No matter, how complex the function $f_i()$ may be, the truth table can be represented by a 4-LUT. Essentially, all the complex expressions for S_i can be expressed in terms of truth table of 4-LUT.
- We can represent the eight output bits for 4 × 4 bits multiplier with eight 4-LUTs.

In general, we can implement a $M \times K$ bit constant multiplication using (M+K) K-input LUTs.

It has been discovered that the LUT size of 4–6 provides the best area-delay product for an FPGA Ahmed and Rose [1]. Most commercial reconfigurable devices such as FPGAs have 4-input LUTs. We therefore discuss the mapping of an $M \times K$ bit constant multiplier into 4-LUTs in the next subsection.

4.1.2 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 $S_1, S_2 \ldots S_{M+K-1}$. 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 implement the multiplication of A and B.

A $(K + 1)$-input LUT can be build from 2 K-input LUTs (as shown in the Figure 9). For example, we can build a 8-LUT from 2 7-LUTs which can be synthesized from $2 \times 2 = 4$ 6-LUTs. Thus, one 8-LUT can be made from $2^4 = 16$ 4-LUTs. Thus, we can build an arbitrary M-LUT from 2^{M-4} 4-LUTs.

Figure 10 gives an example of multiplication of 8-bit number with 12-bit constant ($M = 8$, $K = 12$). Figure 10a gives an implementation using 8-LUTs. 20 8-LUTs or equivalently 128 4-LUTs are used in the design.

Figure 10b gives 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 bits adder. This design requires less LUTs but the presence of 20-bit adder may slow down the clock speed in such a design.

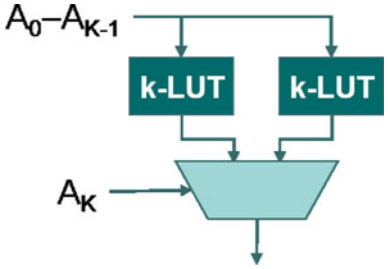

Fig. 9 Building a $(K + 1)$-LUT from K-LUT

Fig. 10 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

4.2 Implementation results

We estimated the hardware performance of our architecture by synthesizing the design on a Xilinx Virtex-4 XC4VLX140 FPGA, using ModelSim SE 6.4 for simulation and Xilinx ISE 10.1 for synthesis. This design just serves as the proof of concept for our architecture. An ASIC implementation with fixed interconnects for LUTs can achieve significant improvements in clock speed and throughputs.

Table 3 shows the performance comparison of SWT architecture with existing works and amongst different various configurations. A direct implementation of Daubechies 9/7 DWT filter requires 16 multiplier and 12 adders in the design. The critical path is $T_m + 4T_a$, where T_m is the time latency of multiplier and T_a is the time latency of adder. Several optimizations were proposed including those by Jou et al. [14]; Vishwanath et al. [33]; Huang et al. [13]. Our earlier work Pande and Zambreno [27, 28] obtains a multiplier-less optimized architecture for DWT that has time latency of only $3T_a$ cycles. On a Virtex-4 FPGA, it obtained a clock frequency of 120 MHz.

A direct implementation of SWT using hardware multipliers gave a clock frequency of 60 MHz. The critical path has one multiplier and five adders ($T_m + 5T_a$). We removed all the multipliers in the design with RCM blocks which reduced the critical path to four adders and one look-up operation ($4T_a + T_l$). (The entire expression for filter coefficients, earlier spanning many multipliers and adders is now represented by a single RCM). The use of reconfigurable multipliers reduces the critical path of the SWT circuit and leads to an improved clock frequency of 114 MHz.

All the reported clock frequency except the VLSI implementation represent implementation on Vitex-4 FPGA. These FPGAs are built on a 90 nm process technology. To test the speed of VLSI implementation of proposed architecture, we used freepdk 45 nm cell library Stine et al. [30]. We were able to target a clock frequency of 500 MHz.

It can easily support HD video at 30 frames per second and resolutions higher than 1440 × 1080.

5 Conclusion and future work

We proposed a DWT design in which the choice of filter coefficients and the orientation of subbands are performed in accordance with a key. The system provides both encryption and security and thwarts brute force attacks. The major contributions of this work are as follows:

1. DWT kernel was parameterized to incorporate the encryption feature and promise reasonable security for real-time embedded multimedia systems.
2. A zero computation overhead subband re-orientation scheme is proposed and implemented in this paper.
3. An optimized hardware implementation of the DWT architecture is presented. The proposed hardware implementation has low critical path and thus achieves a high clock frequency. Reconfigurable hardware based implementation is presented in this paper to embed the key information into the reconfigurable bit stream.

The proposed SWT operation provides a large key-space for multimedia encryption when used as a part of image compression system. As a future work, we propose to parameterize and integrate encryption to other steps in multimedia compression. However, if used by itself, it is prone to cryptanalysis because it retains correlation amongst subbands and some other properties useful in subsequent compression operations.

References

1. Ahmed, E., Rose, J.: The effect of LUT and cluster size on deep-submicron FPGA performance and density. IEEE Trans. VLSI Syst. **12**(3), 288–298 (2004)
2. Baugher, M., McGrew, D., Naslund, M., Carrara, E., Norrman, K.: The secure real-time transport protocol (srtp) (2004)
3. Brachtl, M., Uhl, A., Dietl, W.: Key-dependency for a wavelet-based blind watermarking algorithm. In: Proceedings of ACM workshop on Multimedia and security (MM&Sec) 2004, pp 175–179. ACM, New York (2004). doi:10.1145/1022431.1022462
4. Canvel, B., Hiltgen, A., Vaudenay, S., Vuagnoux, M.: Password interception in a SSL/TLS channel. In: The 23rd Annual International Cryptology Conference, CRYPTO '03, vol 2729, pp 583–599 (2003)
5. Cheng, C.C., Tseng, P.C., Chen, L.G.: Multimode embedded compression codec engine for power-aware video coding system. IEEE Trans. Circuits Syst. Video Technol. **19**(2), 141–150 (2009). doi:10.1109/TCSVT.2008.2009250
6. Cheng, H., Li, X.: Partial encryption of compressed images and videos. IEEE Trans. Signal Process. **48**(8), 2439–2451 (2000). doi:10.1109/78.852023
7. Christopoulos, C., Skodras, A., Ebrahimi, T.: The JPEG2000 still image coding system: an overview. IEEE Trans. Consumer Electr. **46**(4), 1103–1127 (2000)
8. Engel, D., Uhl, A.: Parameterized biorthogonal wavelet lifting for lightweight JPEG 2000 transparent encryption. In: Proceedings of

ACM workshop on Multimedia and security (MM&Sec) 2005, pp 63–70. ACM, New York (2005). doi:10.1145/1073170.1073183
9. FIPS 197.: Announcing the Advanced Encryption Standard (2001)
10. FIPS 46-2.: Announcing the standard for Data Encryption Standard (1993)
11. Grangetto, M., Magli, E., Olmo, G.: Multimedia selective encryption by means of randomized arithmetic coding. IEEE Trans. Multimedia **8**(5), 905–917 (2006)
12. Hodjat, A., Verbauwhede, I.: A 21.54 Gbits/s fully pipelined AES processor on FPGA. In: Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM), pp 308–309 (2004). doi:10.1109/FCCM.2004.1
13. Huang, C., Tseng, P., Chen, L.: Flipping structure: an efficient VLSI architecture for lifting-based discrete wavelet transform. IEEE Trans. Signal Process. **52**(4), 1080–1089 (2004)
14. Jou, J.M., Shiau, Y.H., Liu, C.C.: Efficient VLSI architectures for the biorthogonal wavelet transform by filter bank and lifting scheme. IEEE Int. Symp. Circuits Syst. **2**, 529–532 (2001). doi:10.1109/ISCAS.2001.921124
15. Kim, H., Wen, J., Villasenor, J.: Secure arithmetic coding. IEEE Trans. Signal Process. **55**(5), 2263–2272 (2007). doi:10.1109/TSP.2007.892710
16. Lai, X., Massey, J.L.: A proposal for a new Block Encryption Standard. In: EUROCRYPT '90, pp 389–404. Springer, New York
17. Lian, S., Wang, Z.: Comparison of several wavelet coefficient confusion methods applied in multimedia encryption. In: International Conference on Computer Networks and Mobile Computing, pp 372–376 (2003)
18. Lian, S., Liu, Z., Ren, Z., Wang, H.: Commutative encryption and watermarking in video compression. IEEE Trans. Circuits Syst. Video Technol. **17**(6), 774–778 (2007)
19. Liu, X., Eskicioglu, A.M.: Selective encryption of multimedia content in distribution networks: challenges and new directions. In: Communications, Internet, and Information Technology (CIIT 2003), pp 276–285 (2003)
20. Liu, Z., Zheng, N.: Parametrization construction of biorthogonal wavelet filter banks for image coding. Signal Image Video Process. **1**(1), 63–76 (2007)
21. Mano, M.M., Ciletti, M.D.: Digital Design, 4th edn. Prentice-Hall Inc. Upper Saddle River (2006)
22. Mao, Y., Wu, M.: A joint signal processing and cryptographic approach to multimedia encryption. IEEE Trans. Image Process. **15**(7), 2061–2075 (2006)
23. Marcellin, M., Bilgin, A.: Quantifying the parent-child coding gain in zero-tree-based coders. IEEE Signal Process. Lett. **8**(3), 67–69 (2001). doi:10.1109/97.905942
24. Martin, K., Plataniotis, K.: Privacy protected surveillance using secure visual object coding. IEEE Trans. Circuits Syst. Video Technol. **18**(8), 1152–1162 (2008)
25. Martina, M., Masera, G.: Multiplierless, folded 9/7–5/3 wavelet VLSI architecture. IEEE Trans. Circuits Syst. II **54**(9), 770–774 (2007)
26. Pande, A., Zambreno, J.: An efficient hardware architecture for multimedia encryption and authentication using discrete wavelet transform. In: IEEE CS International Symposium VLSI, pp 85–90 (2009)
27. Pande, A., Zambreno, J.: Poly-dwt: polymorphic wavelet hardware support for dynamic image compression. ACM Trans. Embed. Comput. Syst. (2010a)
28. Pande, A., Zambreno, J.: A reconfigurable architecture for secure multimedia delivery. In: 23rd Intl. Conf. VLSI Design, pp 258–263, (2010b). http://www.computer.org/portal/web/csdl/doi/10.1109/VLSI.Design.2010.50
29. Shannon, C.E.: Communication theory of secrecy systems. Bell Syst. Tech. J. **28**, 656–715 (1949)
30. Stine, J., Castellanos, I., Wood, M., Henson, J., Love, F., Davis, W., Franzon, P., Bucher, M., Basavarajaiah, S., Oh, J., Jenkal, R.: FreePDK: an open-source variation-aware design kit, pp i–iii (2007). doi:10.1109/MSE.2007.3
31. Strang, G., Nguyen, T.: Wavelets and Filter Bank. Wellesley-Cambridge Press, Wellesley (1996)
32. Tseng, P., Chang, Y., Huang, Y., Fang, H., Huang, C., Chen, L.: Advances in hardware architectures for image and video coding—a survey. In: Proceedings of IEEE, vol. 93, issue 1, pp. 184–197 (2005). doi:10.1109/JPROC.2004.839622
33. Vishwanath, M., Owens, R., Irwin, M.: VLSI architectures for the discrete wavelet transform. In: IEEE Trans Circuits and Systems II: Analog and Digital Signal Processing, vol. 42, issue 5, pp 305–316 (1995). doi:10.1109/82.386170
34. Wagner, D., Schneier, B.: Analysis of the SSL 3.0 protocol. In: Proceedings of Second UNIX Work. Electronic Commerce, USENIX Association, pp 29–40 (1996)
35. Wang, S.J., Chen, H.H., Chen, P.Y., Tsai, Y.R.: Security cryptanalysis in high-order improved fast encryption algorithm for multimedia. In: Future Generation Communication and Networking (FGCN 2007), vol 1, pp 328–331 (2007). doi:10.1109/FGCN.2007.199
36. Yi, X., Tan, C.H., Slew, C.K., Rahman Syed, M.: Fast encryption for multimedia. In: IEEE Transactions on Consumer Electronics, vol. 47, issue 1, pp 101–107 (2001). doi:10.1109/30.920426

Author Biographies

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 third 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 with the co-authors 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. He has published over 19 peer-reviewed scholarly research contributions in top international conferences and journals.

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 Ph.D. degree in Electrical and Computer Engineering in 2006, his M.S. degree in Electrical and Computer Engineering in 2002, and his B.S. degree summa cum laude in Computer Engineering in 2001. While at Northwestern University, Dr. Zambreno 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 Ph.D. dissertation titled "Compiler and Architectural Approaches to Software Protection and Security." His research interests include embedded systems, computer architecture and reconfigurable computing, including the design and implementation of optimized FPGA architectures in various real-world domains such as cryptography, image and video processing, and data mining. He has published over 40 peer-reviewed scholarly research contributions in top international conferences and journals.