A Secure Compression Scheme for Real-time Applications Using 2D-WT and Cellular Automata

M. T. Ramírez-Torres, J. S. Murguía, M. Mejía Carlos, and J. A. Aboytes-González

Universidad Autónoma de San Luis Potosí,
Coordinación Académica Región Altiplano Oeste e IICO, San Luis Potosí,
Mexico

tulio.torres@alumnos.uaslp.edu.mx,
ondeleto@uaslp.mx,
marcela.mejia@uaslp.mx,
j.a.a.g.85@hotmail.com
http://salinas.uaslp.mx

Abstract. In this work is presented a numerical implementation of a system that combines a compression scheme with an improved encryption procedure, which is applied to digital images. For the compression stage is considered the two-dimensional Haar wavelet transform, where an energy criterion is contemplated. On the other hand, the encryption scheme regarded is based on the synchronization of the cellular automaton rule 90, this system presents a good performance to encrypt images and it is resistant to cryptanalysis attacks such as the Chosen/Known-plaintext attack. The numerical conjunction of these procedures could be an appealing option for real-time applications such as video communications, video-surveillance among others.

Keywords: Cellular automata, encryption system, two-dimensional wavelet transform, compression

1 Introduction

Nowadays, there exists a great interest in the protection and manipulation of the data. Due to the great advances in technology, each time is required to have better and more efficient algorithms for confidential and secure data handling. This information may vary depending on the application area and in many cases is necessary processing it in real time. For example, now is very common in different countries that the police install surveillance video cameras on the cities. If the information transmitted from these cameras is not encrypted the confidentiality of data is exposed in the links allowing access to third parties without being detected. But if an encryption processes is added, the latency of the data transmission could increase and the information will not be available on time.

To overcome the eavesdropping problem several encryption systems have been proposed, such as AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), RSA (Rivest, Shamir y Adleman) among others. These systems are generally used in text and binary data, but they are not suitable for the encryption of multimedia data due to their massive volumes, high adjacent correlation and sometimes the multimedia data require real-time interactions (displaying, bit rate conversion, etc.) [2]. Hence many encryption systems with different approaches have been developed for image encryption area [5,8].

On the other hand, different compression schemes which can either be lossless or lossy, work by squeezing redundancy out of data, reducing substantially the initial size of the analysed signals. In this issue, the wavelet transform has proved to be a powerful tool to efficiently process signals that involve large amounts of information. In particular, it has been noticed that this transform is a flexible mathematical tool employed in a great variety of applications and its numerical implementation is often easy to perform [3].

The encryption system considered in this implementation is based on the synchronization of the cellular automaton rule 90 [9]. This cryptosystem is named ESCA (for short) and it has been validated and implemented for image encryption in [6,7]. The ESCA system can be considered secure for images encryption but latency time is too high to incorporate it in real-time applications. In this work we implement a joint encryption and compression procedure to image information. The compression scheme is based on the Haar wavelet transform with an energy approach. The results show that this scheme could be an efficient solution to protect information with a low latency. The structure of this paper is organized as follows. Section 2 discusses briefly the two-dimensional Haar wavelet transform as a tool to compress images. The encryption system and the image encryption algorithm are described in Section 3. The numerical implementation of the joint scheme and results of this proposal is discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2 Two-Dimensional Wavelet Transform

The discrete wavelet transform has a huge number of applications in different areas, and many signals have a bi-dimensional nature like images. For this case, the wavelet transform has also a discrete version to process them. The wavelet transform in two dimensions considers a two-dimensional scaling function, $\Phi(x, y)$, and three two-dimensional wavelets, $\Psi^H(x, y)$, $\Psi^V(x, y)$, and $\Psi^D(x, y)$, where the superscript index indicates the information of the signal at the horizontal (H), vertical (V), and diagonal (D) directions. Each function corresponds to the product of a one-dimensional scaling function φ and corresponding wavelet ψ such that the product does not produce a one-dimensional result, i. e.,

$$\Phi(x,y) = \varphi(x)\varphi(y), \tag{1}$$
$$\Psi^H(x,y) = \psi(x)\varphi(y), \tag{2}$$
$$\Psi^V(x,y) = \varphi(x)\psi(y), \tag{3}$$
$$\Psi^D(x,y) = \psi(x)\psi(y)\ . \tag{4}$$

Given separable two-dimensional scaling and wavelet functions, we define the scale and translation versions as:

$$\Phi_{j;m,n}(x,y) = 2^{j/2}\Phi(2^j x - m, 2^j y - n), \tag{5}$$
$$\Psi^d_{j;m,n}(x,y) = 2^{j/2}\Psi^d(2^j x - m, 2^j y - n), \tag{6}$$

where $j, m, n \in Z$ and the superscript index d assumes the values H, V and D to identify the directional wavelets given in (2)-(4).

In the same spirit as in the case of the DWT in one dimension, and considering that (5)-(6) constitute an orthonormal basis for $L^2(\mathbb{R}^2)$, the expansion of a function $f(x,y)$ of finite energy is then

$$\begin{aligned} f(x,y) = \frac{1}{\sqrt{UV}} \sum_m \sum_n \mathbf{a}_{j_0;m,n}\Phi_{j_0;m,n}(x,y) \\ + \frac{1}{\sqrt{UV}} \sum_{d=H,V,D} \sum_{j=j_0} \sum_m \sum_n \mathbf{d}^d_{j;m,n}\Psi^d_{j;m,n}(x,y), \end{aligned} \tag{7}$$

where the scaling $\mathbf{a}_{j;m,n}$ and wavelet $\mathbf{d}^d_{j;m,n}$ coefficients are defined as

$$\begin{aligned} \mathbf{a}_{j;m,n} &= \int\int f(x,y), \Phi_{j;m,n}(x,y)dxdy, \\ \mathbf{d}^d_{j;m,n} &= \int\int f(x,y), \Psi^d_{j;m,n}(x,y)dxdy\ . \end{aligned} \tag{8}$$

Equation (7) represents the synthesis equation, whereas (8) is the analysis equation. Both equations constitute the two-dimensional discrete wavelet transform (2D-DWT). From now on, unless otherwise stated, we refer a two-dimensional function or signal $f(x,y)$ as an image function $\mathbf{I}(x,y)$ with dimensions $U \times V$, since the 2D-DWT is generally used to image analysis.

To compute numerically the two-dimensional wavelet transform, we follow the Mallat's algorithm for two-dimensional functions [3]. With this algorithm, the multiresolution decomposition of a two-dimensional function or an image is represented by a series of approximations and details of sub-images, which become increasingly coarse. In general, a 2D-DWT can be considered as a separable filter bank of row and column directions that decomposes one resolution

level of an image into four sub-images. A one stage of this procedure is shown in Figure 1, where h and g correspond to a lowpass and highpass filter, respectively, and they are followed for the operation of downsampling by two. After applying the first wavelet level transform, we have four sub-images, and if the original image function $\mathbf{I}(x, y)$ has dimensions $U \times V$, then each sub-image have $U/2$ rows and $V/2$ columns. The approximation sub-image is obtained by computing approximations along rows of the signal $\mathbf{I}(x, y)$ followed by computing approximations along columns. This sub-image is an averaged version of the image $\mathbf{I}(x, y)$ with half resolution and with statistical properties that are similar to those of the original signal $\mathbf{I}(x, y)$. In the same way, in the horizontal sub-image, we first compute the approximations along the rows of the image $\mathbf{I}(x, y)$ followed by computing the details along the columns. As a result, the horizontal edges of $\mathbf{I}(x, y)$ will be always detected by the details along the columns. Since this sub-image analyses the horizontal information, it is clear why it is denoted as the horizontal sub-image. In the multiresolution decomposition the same wavelet transformation is applied but only to the approximation sub-image obtaining again four sub-images, but now with dimensions of $U/2^k$ rows and $V/2^k$ columns, where $k = 1, \ldots, \min(\log_2(U), \log_2(V))$ is the wavelet level. The two-dimensional Haar wavelet transform is considered in this paper, because with this wavelet function the algorithm is memory efficient and reversible.

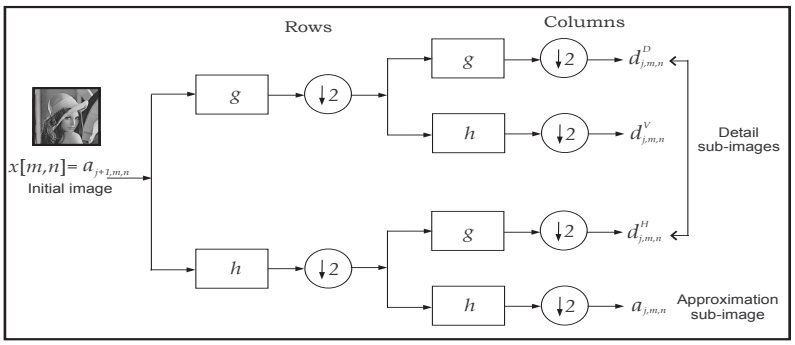

Fig. 1. One stage in a multiresolution image decomposition.

Wavelet compression scheme. The compression procedure that we employ is shown in Figure 2. First of all, the Haar wavelet transform is applied to the initial image. Next, the transformed image is submitted to an elimination process of the transformed coefficients which lie below a threshold value. In fact, the key step here is to choose a threshold through an energy criterion. We look at the normalized cumulative energy applied to the ordered transformed coefficients. To select the threshold value we consider the magnitude of the coefficient for which a proposed energy percentage is obtained. With an established threshold

value ε any coefficient in the wavelet transformed data whose magnitude is less than ε will be reset to zero. Hence the amount of obtained compression can be controlled by varying the threshold parameter ε.

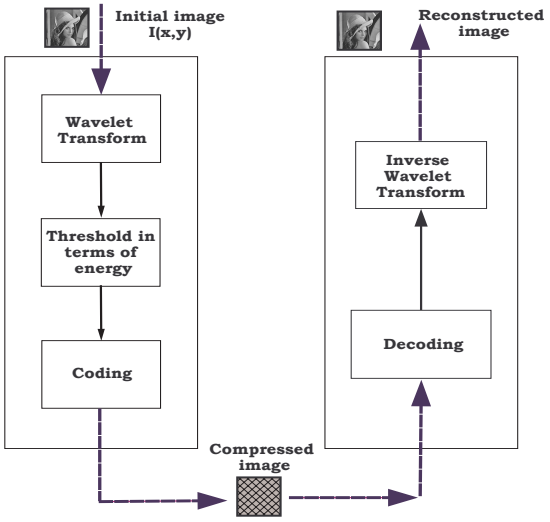

Fig. 2. Basic wavelet compression procedure with an energy approach.

3 Encryption System ESCA

In this work is considered the encryption scheme used in [9], where the synchronization phenomenon of cellular automata has been applied to design two families of permutations Ψ and Φ, and an asymptotically perfect pseudo-random number generator. This cryptosystem is flexible and reconfigurable for different bit-lengths. Figure 3 illustrates a general block diagram of the encryption system ESCA.

The ESCA system comprises the sets M, C and K of binary words, M and C correspond to the plaintexts and ciphertexts respectively, the length of these words is $J = 2^j$, for $j = 1, 2, 3....$ Also it is possible to concatenate several blocks of these lengths. The set K corresponds to the enciphering keys of length $N = 2^n - 1$ for $n = 1, 2, 3...$, for a complete encryption $n > j$ such that N must be larger than J. The two indexed families of permutations $\Psi = \{\psi_{\mathbf{k}} : \mathbf{k} \epsilon K\}$ and $\Phi = \{\phi_{\mathbf{k}} : \mathbf{k} \epsilon K\}$ are called encryption and decryption functions respectively. Basically, the cryptosystem transforms a plaintext sequence $\mathbf{m}$ into a ciphertext sequence $\mathbf{c}$, i.e. for every $\mathbf{k} \epsilon K$ one has $\mathbf{c} = \psi_{\mathbf{k}}(\mathbf{m})$, whereas to disclose from the sequence of cipher-blocks, one uses the decryption function $\mathbf{m} = \phi_{\mathbf{k}}(\psi_{\mathbf{k}}(\mathbf{m}))$.

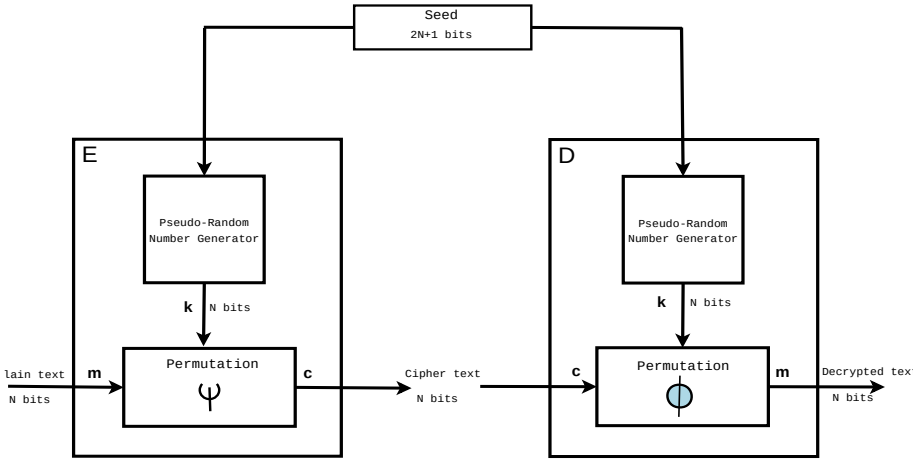

Fig. 3. The encryption scheme ESCA with its main components: the indexed families of permutations and the pseudorandom generator keys.

Since the complete encryption scheme is a symmetric algorithm, the encryption and decryption processes use the same enciphering key $\mathbf{k}$.

In [4], the authors present an ergodic and mixing transformation of binary sequences in terms of a cellular automaton, which is the main element of a pseudo-random number generator (PRNG). The PRNG in its basic form, follows the algorithm shown in Figure 4. At first, the key generator requires two seeds, $\mathbf{x} = \mathbf{x}_0^{k+1}$, of N bits, and $\mathbf{y} = \mathbf{x}_0^k$, of $(N+1)$ bits, which are the input of function $\mathbf{k} = h(\mathbf{x}, \mathbf{y})$. The seeds are $\mathbf{x} = \{x_1, x_2, x_3, \ldots, x_N\}$ and $\mathbf{y} = \{y_1, y_2, y_3, \ldots, y_{N+1}\}$, and the first number generated of N bits is the sequence output of function h, $\mathbf{k} = x_0^1 = \{k_1, k_2, k_3, \ldots, k_N\}$. Now this sequence is feeding back to the input, which becomes the next value of $\mathbf{x}$, and the previous value of $\mathbf{x}$ becomes the initial bits of the new $\mathbf{y}$, where the missing bit is the least significant bit (LSB) of the previous $\mathbf{y}$, which becomes the most significant bit (MSB) of this sequence, and the same procedure is iterated repeatedly.

In [6], it was added a pre-processing to make this scheme resistant to Chosen / Known-plaintext attacks. This process is similar to PRNG, where the function $\widehat{\mathbf{m}} = h(\mathbf{m}, \mathbf{z})$ is used to transform the blocks $\mathbf{m}$ into an unintelligible form denominated $\widehat{\mathbf{m}}$. The Figure 5 shows a block diagram of this process, the inputs are a block $\mathbf{m}$ of J bits and a seed $\mathbf{z}$ of $J+1$ bits. At the output is obtained a block $\widehat{\mathbf{m}}$ of J bits, that will be encrypted later with the permutation Ψ, $\mathbf{c} = \psi_{\mathbf{k}}(\widehat{\mathbf{m}})$. Also in the Figure 5 is shown the feedback, this is different from the key generation, the next $\mathbf{z}$ is obtained from the joint of $\widehat{\mathbf{m}}$ and the LSB of previous $\mathbf{z}$ as the MSB. This change allows to process images with a high adjacent correlation.

The encryption of an image with the ESCA system proceeds as follows

1. Load the plain-image $\mathbf{I}$ of size $V \times U$.

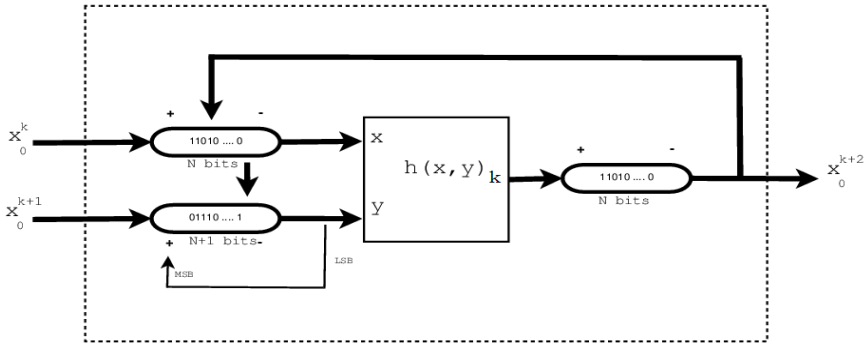

Fig. 4. Basic form of the pseudo-random number generator.

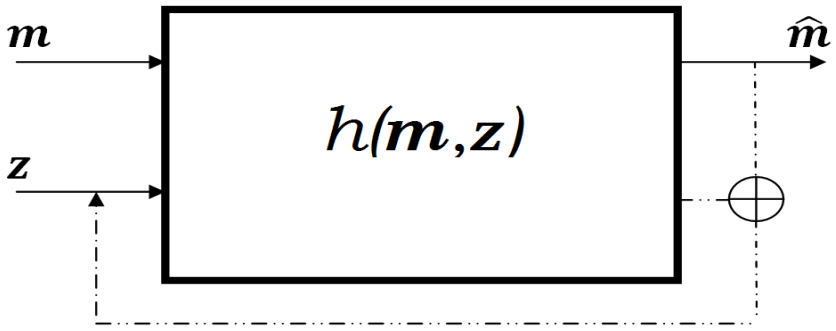

Fig. 5. Pre-processing to obtain $\widehat{\mathbf{m}}$.

2. By scanning the image I row by row, arrange its respective pixels as a sequence or a vector, and convert each pixel value to their corresponding binary value.
3. Establish the length of the encryption key, it must be larger than plaintext bit-length.
4. Compute the modified plaintext sequence $\widehat{\mathbf{m}}$ using the pre-processing.
5. Encrypt each modified block, $\mathbf{c} = \psi_{\mathbf{k}}(\widehat{\mathbf{m}})$ with a different $\mathbf{k}$ each one.
6. By reshaping the set of ciphered sequences of the previous step into an $V \times U$ image, obtain the ciphered image.

This algorithm has proved to be secure against statistical attacks and model threats as Chosen/Known plaintext attacks [6] [7].

4 Numerical Implementation and Results

To implement numerically the joint scheme, it was considered the LabVIEW graphical programming language, a trademark of National Instruments [1]. In Figure 6 is depicted this scheme, which comprises two stages. The top block, Module A, carries out the compression and encryption of images, as was described above, whereas the bottom block, Module B, performs the reverse process to obtain a reconstructed image. It is worth to say that there are two signals after the compression stage, the wavelet coefficients that survived to the value of the threshold, which are the values to encrypt, and a binary vector indicating the positions of such coefficients. Hence the encrypted image and the binary vector are available to be transmitted through a public channel, but it will depend on the application if it is required to convey. Of course, in the Module B the received image is decrypted, and the decompression procedure takes place to the decrypted image with the binary position vector obtaining a reconstructed image.

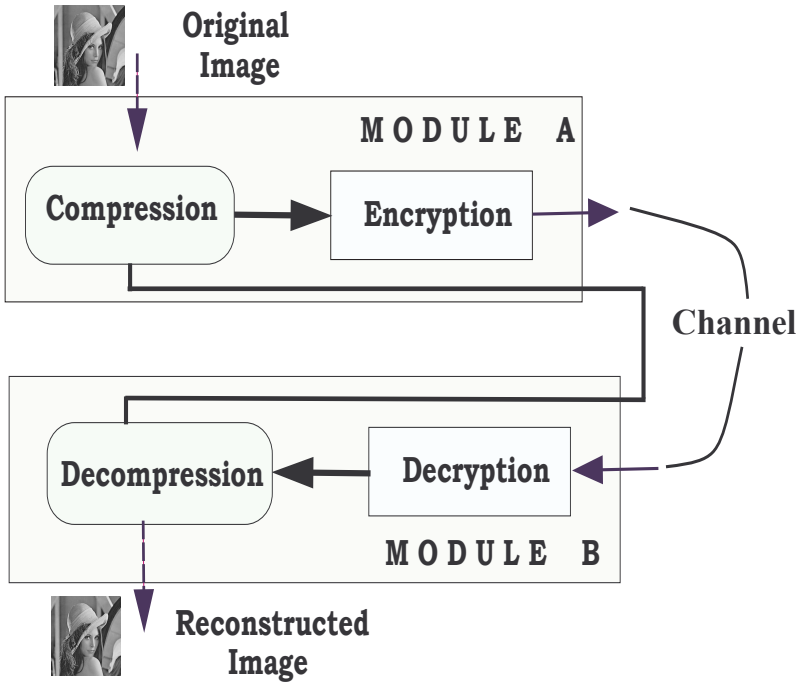

Fig. 6. Image compression encryption scheme.

In such numerical implementation the system was tested with 512×512 RGB and grayscale images, Lena, mandrill and peppers. For the RGB versions, the compression stage should be applied for each color channel. To measure the

degradation of the reconstructed images it was used Peak Signal-to-Noise Ratio (PSNR) as a quality metric. The individual results of the wavelet compression procedure for the Lena image can be observed in Figure 7, it illustrates some reconstructed images for different energy criteria and its PSNR. In Table 1 is shown the compression rate of the RGB images with a 50% of energy. Based on these results is clear that the Haar wavelet transform helps to achieve good compression rates, and the PSNR helps to determine degradation on the reconstructed images.

Fig. 7. a) The source image. Reconstructed images when a b) 90% (37.32 dB), c) 75% (30.32 dB), and d) 50% (23.70 dB) of energy is considered in the compression stage.

Table 1. Compression rate achieved with 50% of energy and 4 levels. PSNR of the reconstructed images from these settings.

Image	**Compression rate**	**PSNR**
Lena	24.68:1	24.68 dB
mandrill	23.93:1	18.60 dB
peppers	38.34:1	34.38 dB

The ESCA system encrypts the survived coefficients, for RGB images before to encrypt the survived coefficients from each channel, they are concatenated again. Figure 8 illustrates the results obtained in the stages contained in Module A for a source Lena image. In this case, an energy criterion of 90% was considered to compress the source image.

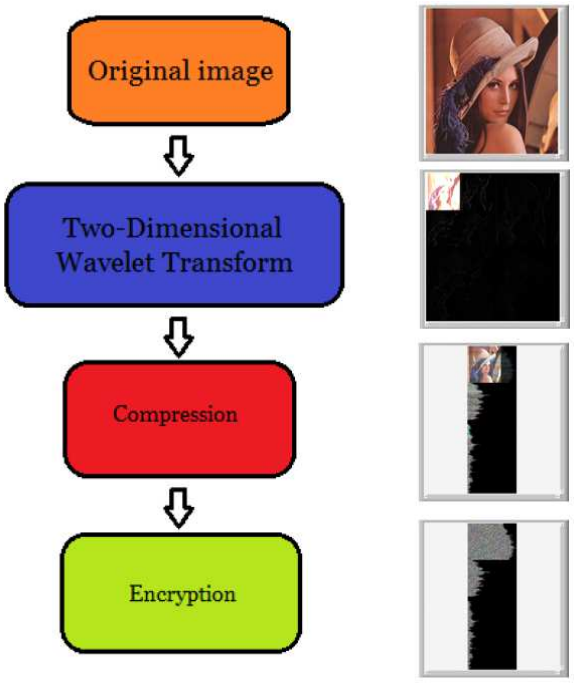

Fig. 8. Results of the application of Module A to a source Lena image.

The results of this scheme show that is possible to reduce latency time until 80% in comparison with only encryption scheme. The Table 2 contains a comparative between latency time (in milliseconds) of the encryption process without compression stage and our proposed scheme, using RGB images with a resolution of 128×128 pixels, considering a 75% of energy and 4 levels of 2D-WT. The Table 2 also shows the PSNR of the recovered images and the compression ratio reached.

Finally, this scheme was implemented in a video application using a conventional webcam with a resolution 128×128 pixels and 30 frames per second. In this program the user can select the number of levels of the two-dimensional wavelet transform and the energy percent to preserve. In the same front panel is shown the compressed and encrypted images as result of Module A, and the recovered images as output of the Module B. Also the program calculates the PSNR of the recovered video, in this way the user can select the parameters

Table 2. Comparison of latency times between different schemes.

	Encryption process (ms)	Our proposed scheme (ms)	PSNR (dB)	Compression rate
Lena	72	28	24.07	11:1
mandrill	72	29	21.54	10:1
peppers	72	26	23.72	12.5:1

according to the quality of the recovered image and the latency based on the flow of the recovered video. The Figure 9 shows the front panel of the video application.

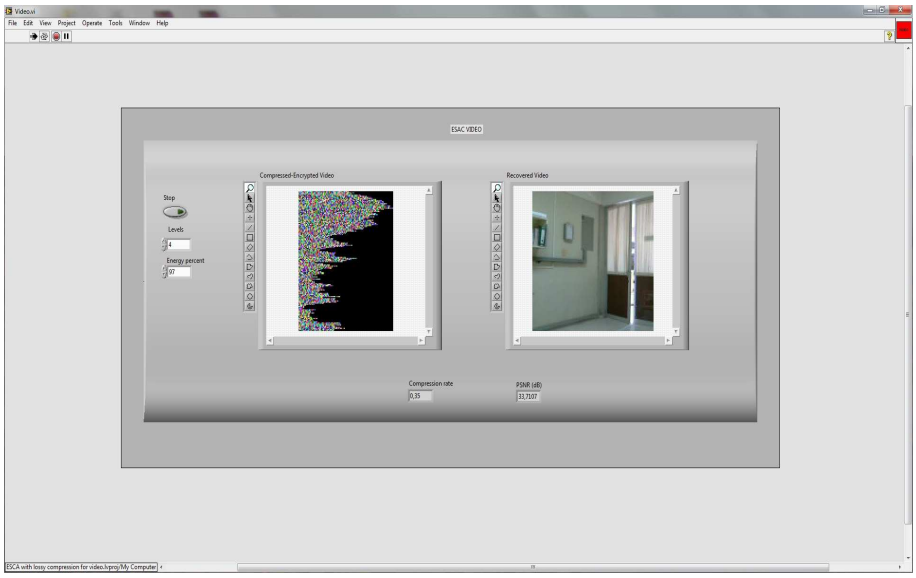

Fig. 9. Front panel of the video application with the proposed scheme, considering 97% of energy, 4 levels and the recovered images have a is 33.71 dB.

5 Conclusions

In this work we presented the numerical implementation of a system that combines a compression procedure with an improved encryption system, which was applied to images. The compression procedure is based on an energy criterion of the Haar wavelet coefficients, and the obtained results provide us good compression rates, because a high energy concentration was presented in a few wavelet

coefficients of the transformed image. On the other hand, the improvement of the encryption system presented a high security and a great flexibility to encrypt image information. In fact, this allowed that some cryptanalysis attacks were outperformed, and now with the compression stage it presented a good performance in time efficiency issues. With the obtained results of this proposal, we think that it can be a useful tool in the current multimedia applications.

Acknowledgments. J. A Aboytes-González is doctoral fellow of CONACYT (México) in the program of "Ciencias Aplicadas" at IICO-UASLP. The authors also want to thank to FAI of UASLP for the economical support to this work.

References

1. The labview environment. url http://www.ni.com/labview/
2. Lian, S.: Multimedia content encryption: techniques and applications. CRC press (2008)
3. Mallat, S.: A wavelet tour of signal processing. Academic press (1999)
4. Mejıa, M., Urıas, J.: An asymptotically perfect pseudorandom generator. Discrete and Continuous Dynamical Sys 7, 115–126 (2001)
5. Pareek, N.K., Patidar, V., Sud, K.K.: Image encryption using chaotic logistic map. Image and Vision Computing 24(9), 926–934 (2006)
6. Ramírez-Torres, M., Murguía, J., Carlos, M.M.: Image encryption with an improved cryptosystem based on a matrix approach. International Journal of Modern Physics C 25(10), 1450054 (2014)
7. Ramirez-Torres, M., Murguia, J., Mejia-Carlos, M.: Fpga implementation of a reconfigurable image encryption system. In: ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on. pp. 1–4. IEEE (2014)
8. Suresh, V., Madhavan, C.V.: Image encryption with space-filling curves. Defence Science Journal 62(1), 46–50 (2012)
9. Urıas, J., Salazar, G., Ugalde, E.: Synchronization of cellular automaton pairs. Chaos: An Interdisciplinary Journal of Nonlinear Science 8(4), 814–818 (1998)