# VIDEO ENCRYPTION WITH CHAOTICALLY COUPLED CHAOTIC MAPS USING DMP-6437 AND SIMULINK

R. JAIMES-REATEGUI[1], EMMANUEL VALDES JARAMILLO[1], RICARDO SEVILLA-ESCOBOZA[1], JUAN HUGO GARCÍA LOPEZ[1], CARLOS E. CASTAÑEDA HERNANDEZ[1], GUILLERMO HUERTA-CUELLAR[1], AND D. LOPEZ-MANCILLA[1]

[1]Centro Universitario de los Lagos, Universidad de Guadalajara, Enrique Díaz de León s/n. Lagos de Moreno, Jal., C. P. 47460, México,

MASSIMILIANO ZANIN[2] AND ALEXANDER N. PISARCHIK[3]

[2] The Innaxis Research Institute, Velázquez 157-Ibercenter, 28002 Madrid, Spain

[3]Centro de Investigaciones en Óptica, Lomas del Bosque 115, León, Gto., C. P. 37150, México.

rjaimes@cio.mx,

*ABSTRACT. To encrypt video we use a secure cryptosystem for direct encryption of color images in each frame of a video, based on chaotically coupled chaotic maps, that provides good confusion and diffusion properties that ensures extremely high security because of the chaotic mixing of pixels colors, using a DMP (Digital Media Processor) we process video, separate it in 3 components RGB (Red, Green, Blue) and apply our algorithm for encryption or decryption.*

**Keywords:** Chaotic maps, video encryption, DMP 6437.
**Paper ID RJI272**

## 1. INTRODUCTION

In recent years the security of communications has been an important issue for companies. The implementation of encryption algorithms that are slow to encrypt video has complicated the scene. The problem of security in storage and transmission of confidential visual information is therefore growing in importance, and requires solutions for many applications, such as pay TV, video conferencing, medical and military databases. Most conventional ciphers, such as Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES), linear feedback shift register (LFSR), etc. [1,2] with high computational security consider plaintext as either block cipher or data stream and are not suitable for image/video encryption in real time because their speed is slow due to a large data volume and strong correlation among image pixels. The implementation of traditional algorithms for image encryption is even more complicated when undertaken with commercial software.

In the last decades many researchers pointed out the existence of a strong relation between chaos and cryptography [3-5]. The greatest advantage of a chaotic system over a noisy one is that the chaotic system is deterministic. This property of chaos significantly facilitates the decryption process. The exact knowledge of initial conditions and system parameters enables one to recover a message. Moreover, many fundamental characteristics of chaos, such as a broadband spectrum, ergodicity and high sensitivity to initial conditions are directly connected with two basic properties of good ciphers: confusion and diffusion.

This paper is organized as follows. In Sec. 2 chaotic cryptosystem, we describe our algorithm in detail. In the Sec. 3-Sec. 5, we apply it for encryption of a real color image and analyze its security. Finally, the main conclusions are given in Sec. 6.

## 2. CHAOTIC CRYPTOSYSTEM

Any digital image can be represented as a linear array of decimal values, where each pixel has three color components pR, pG, and pB (red, green and blue) of integers between 0 and 255. Since video is 30 frames per second we can process each image of a frame.

An M × N pixel image can be transformed into an array of h = 3*M*N items of the color components that forms plaintext

$$P = \{p_1 = p_1^R,\ p_2 = p_1^G, p_3 = p_1^B, p_4 = p_2^R, \ldots, p_i, \ldots, p_h = p_{MN}^B\}. \qquad (1)$$

The color components of an 8-bit RGB image are integers in the range [0, 255], rather than floating-point values in the range [0, 1]. As we already mentioned in the introduction, for shorter EDT the length of ciphertext should be equal to the length of plaintext. The aim of our encryption algorithm is to create ciphertext.

$$R = \{r_1, r_2, \ldots, r_{i,\ldots,} r_h\} \qquad (2)$$

which contains h encrypted color components $r_i \in$[0, 255]. It's known that the security of a cryptosystem is determined by its confusion and diffusion properties and its sensitivity to secret keys. The basic idea underlying our encryption algorithm is to use chaotic systems for both the confusion and diffusion processes.

First, plaintext P is converted into an array of floating - point values:

$$X_0 = \left\{x_0^{(1)}, x_0^{(2)}, \ldots, x_0^{(i)}, \ldots, x_0^{(h)}\right\} \qquad (3)$$

That can serve as initial conditions for chaotic maps for example the logistic map:

$$x_{n+1} = ax_n(1 - x_n) \qquad (4)$$

Which is chaotic when $a \in$ [3.57, 4]². Both the parameter $a$ and the number of iterations n serve as secret keys. To transform P into X0, we convert the color components to the map variables so that they lie within the chaotic attractor of the map Eq. (4), i.e., the normalized value

$$X_0 = P/[255(x^{max} - x^{min})] \qquad (5)$$

Where $x^{max}$ and $x^{min}$ are the maximum and minimum values of x generated by the chaotic map Eq. (4).

## 3. CHAOTIC COUPLING

In the algorithm, all chaotic maps representing the pixels' values were coupled unidirectionally so that $x_n^i$ was used as the initial condition for the subsequent map (i + 1). Such a coupling provides diffusion properties so good that after three rounds, the cryptosystem becomes extremely sensitive to any change in a pixel value and a small error in one only pixel diffuses over the whole

## 4. FIGURES

This section shows the encryption process for black and white images Figure1 (a) and colour images Figure1 (b), “I” means the rounds that the algorithm applies to image.

Figure 2 shows the encryption / decryption in real time colour video, in this case the process is performed to the YCbCr components, that are native to DMP (Digital Media Processor) card and simplifies the process of encrypting / decryption.

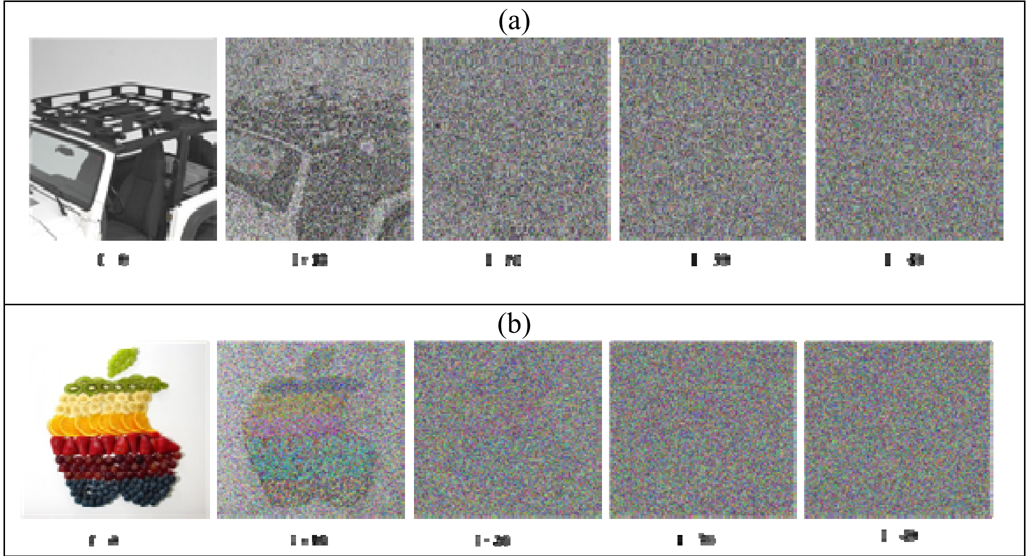

**Figure 1.** (a) Black and white image. (b) Colour image

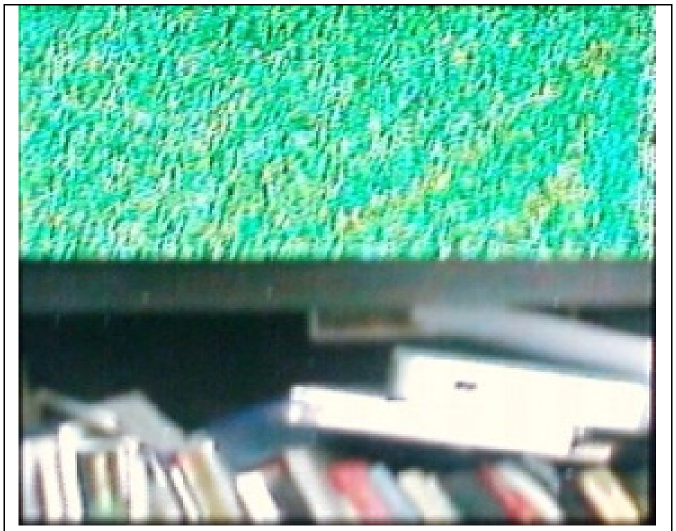

Figure 2 Encryption / decryption in real time colour video

In Figure 3 shows the flow chart in simulink where each letter indicate the process on the card DMP (Digital Media Processor) that was performed to encrypt the video. Each process is detailed below:

A) Capture an NTSC/PAL video input and make it available as a stream of YCbCr 4:2:2 interleaved data.

B) This block separates interleaved YCbCr 4:2:2 data into its luma component (Y'), blue -difference chroma component (Cb), and red-difference chroma component (Cr).

C) The Resize block enlarges or shrinks an image by resizing the image along one dimension (row or column). Then, it resizes the image along the other dimension (column or row).

D) The Color Space Conversion block converts color information between color spaces, in this case we convert YCbCr to RGB.

E) The Logistic Map block process and encrypth the information.

F) The Color Space Conversion block converts color information between color spaces, in this case we return RGB to YCbCr.

G) This block takes planar YCbCr 4:2:2 data on three separate inputs and converts them to a single interleaved YCbCr 4:2:2 data output.

H) This block configures the Video Processing Back End (VPBE) to display NTSC/PAL video Dialog Box.

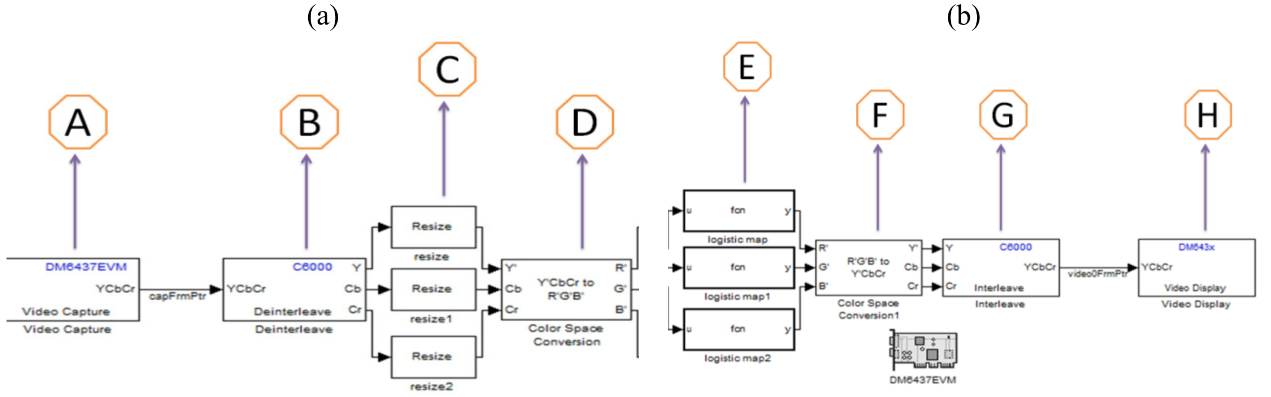

Figure 3. Simulink flow chart (a) part 1. (b) part 2.

The logistic map equation (4) which exhibits some sort of chaotic behavior, when “**a**” Є [3.57, 4]. The parameter “**a**” and the number of iterations “**n**” serve as secret keys.

## 5. PROGRAM CODE

Program code showing two steps in the algorithm showing how the chaotic values are generated and how those values are mixed with the RGB image values.

```
%CONFUSSION STEP%
tmez=zeros(1,cntmax);
tini=0.295173654;
tmez(1)=tini;
mu=3.9371456;
for n=1:cntmax-1
    tmez(n+1)=mu*tmez(n)*(1-tmez(n));
end
```

```
% DIFUSSION STEP
K1 = round( tmez * ( cntmax-1 ) ) + 1;
K2 = mod(((1:cntmax) - K1 + cntmax), cntmax) + 1;
for j = 1:cntmax
  tdifR = img_finRD( K2(j) );
  tdifG = img_finGD( K2(j) );
  tdifB = img_finBD( K2(j) );
for n = 1:10
     tdifR = mu * tdifR * (1 - tdifR);
     tdifG = mu * tdifG * (1 - tdifG);
     tdifB = mu * tdifB * (1 - tdifB);
  end
  tdifR = round(tdifR * 256.0) / 256.0;
  tdifG = round(tdifG * 256.0) / 256.0;
  tdifB = round(tdifB * 256.0) / 256.0;
  img_finRD(j) = mod((img_finRD(j) + tdifR), 1);
  img_finGD(j) = mod((img_finGD(j) + tdifG), 1);
  img_finBD(j) = mod((img_finBD(j) + tdifB), 1);
  img_finRD(j) = round(img_finRD(j) * 256.0) / 256.0;
  img_finGD(j) = round(img_finGD(j) * 256.0) / 256.0;
  img_finBD(j) = round(img_finBD(j) * 256.0) / 256.0;
end
```

## 6. CONCLUSIONS

We could demonstrate that the use of coupled chaotic maps to encryption is a secure system and due to the high speed, the proposed cryptosystem is suitable for application in real-time communication systems like video encryption.

## ACKNOWLEDGEMENTS

This work was supported in part by programa investigacion temprana, folio 2009-700, fondo coecytjal – udeg 2009-1 and sistema estatal de investigadores sei-2009

## REFERENCES

[1]. B. Schneier, Applied Cryptography—Protocols, Algorithms, and Source Code, second ed., C. John Wiley & Sons, Inc., New York, 1996.

[2]. J. Daemen, B. Sand, V. Rijmen, The Design of Rijndael: AES — TheAdvanced Encryption Standard, Springer-Verlag, Berlin, 2002.

[3]. Massimiliano Zanin, Alexander N. Pisarchik Physica D 237 (2008) 2638–2648. Image encryption with chaotically coupled chaotic maps.

[4]. S.G. Lian, Sun, Z. Wang, Physica A 351 (2005) 645-661. Security analysis of chaos-based image encryption algorithm.

[5]. [3] A.N. Pisarchik, N.J. Flores-Carmona, M. Carpio-Valadez, Chaos 16 (3) (2006), Encryption and decryption of images with chaotic map lattices.