People's Demecratic Republic of Algeria
Ministry of Higher Education and Scientific research
M'hamed Bougara University, Boumerdes
Institute of Electrical and Electronic Engineering,
Laboratory of Signals and Systems (LSS)

ALGERIAN JOURNAL OF SIGNALS AND SYSTEMS

ISSN : 2543-3792

Title : **Hyperchaos-Based Cryptosystem for Multimedia Data Security**

Authors: **S. Benzegane, S. Sadoudi, M. Djeddou**
Affiliation: **Laboratoire Systèmes de Communications, Ecole Militaire Polytechnique(EMP), Algiers, Algeria.**

Page range: **47- 58**

Volume : 1 Issue : 1 (June 2016)

Laboratory of Signals and Systems
Address : IGEE (Ex-INELEC), Boumerdes University, Avenue de l'indépendance, 35000, Boumerdes, Algeria
Phone/Fax : 024 79 57 66
Email : lss@univ-boumerdes.dz ; ajsyssig@gmail.com

Hyperchaos-Based Cryptosystem for Multimedia Data Security

S. Benzegane, S. Sadoudi* and M. Djeddou
Laboratoire Systèmes de Communications, Ecole Militaire Polytechnique(EMP), Algiers, Algeria
*sadoudi.said@gmail.com

Abstract: In this paper, we present a software development of multimedia streaming encryption using Hyperchaos-based Random Number Generator (HRNG) implemented in C#. The software implements and uses the proposed HRNG to generate keystream for encrypting and decrypting real-time multimedia data. The used HRNG consists of Hyperchaos Lorenz system which produces four signal outputs taken as encryption keys. The generated keys are characterized by high quality randomness which is confirmed by passing standard NIST statistical tests. Security analysis of the proposed encryption scheme through image and audio security analysis confirms its robustness against different kind of attacks.

Keywords: Hyperchaos Lorenz System, HRNG, Multimedia security, C#.

1. INTRODUCTION

In recent years, with the rapid growth of wireless multimedia service, there is an increasing requirement for higher secure data transmission such as video and audio data. In this way, to protect multimedia contents, cryptology, which appears to be an effective way for information security, has been employed in many practical applications [1]. However, traditional ciphers like DES [2], IDEA [3], RSA [4] and AES [5], are often used for text or binary data, while not suitable for direct video and audio encryption [6]. The main reason is that multimedia applications require real-time operations, which imposes logically the use of stream ciphers instead of block ciphers.

Recently, an increasing attention has been devoted to the usage of chaos theory to implement encryption process. The main advantage of these encryptions lies in the observation that a chaotic signal looks like noise for non-authorized users ignoring the mechanism for generating it [7-13]. In fact, chaotic system is sensitive to initial condition values, this means that the different initial conditions produce different trajectories, but the same conditions can produce the same trajectories. It is known that most chaotic cryptosystems in essence behave as stream ciphers [12]. In this way, the authors in [10] and [11] have proposed hardware chaos-based stream cipher for ciphering audio and image data respectively. The cryptosystems are implemented in FPGA circuits which offer high speed and low cost. However, applying these hardware chaos-based cryptosystems to secure real-time multimedia data seem to be very difficult. For that reason, and with the rapid development of Internet and the place taken by multimedia data exchange in internet, software chaos-based cryptosystems appear as an interesting and less difficult alternative to secure multimedia data instead of hardware cryptosystems. In fact, many research works are proposed in this domain. We cite the work in [13], where a new fast and light stream cipher, named Enigmedia, based on a hyperchaotic dynamical system is proposed. It has been implemented in a videoconference Application for Smartphone. In [14-16] various cryptographic schemes are reported for streamcipher and image encryption. In addition, many chaotic Map based block ciphers are proposed including [17-18]. However, in is known that 1D or 2D chaotic Map systems have a poor chaotic dynamical behavior. In [6], the authors give an interesting survey of chaos-based encryption algorithms for image, video and audio respectively. They affirm that chaos-based multimedia encryption demonstrates superiority over the conventional encryption methods and can be used as the foundation of future research. However, they suggest that chaos-based multimedia encryption is not yet mature and more efforts are needed for its further development toward practical applications with high security, low computational complexity, invariance of compression ratio, format compliance, real-time, multiple levels of security, and strong transmission error tolerance.

In this paper, we propose a software development application designed in C# using the ".NET Framework" for real-time multimedia data encryption in Wi-Fi network. We exploit the four simultaneously generated signals of the implemented hyperchaotic Lorenz system to generate cryptographic keys with good statistical properties. To implement the Lorenz system, we use the well-known fourth order Runge-Kutta (RK-4) method for resolving autonomous continuous chaotic system models. For encoding data, we use base64 encoded numbers class provided with the ".NET Framework" which use 64 bits (16Q48) double precision data format after removing the decimal parts, i.e, we take only the fractional parts. The generated keys are validated through the NIST statistical tests [16], which demonstrate their good statistical performances. In addition, a security analysis of both, encrypted image and encrypted audio results, are realized by analyzing histogram of ciphered data, key sensitivity of the algorithms, statistical analysis using the correlation parameters and entropy and by looking at the key space, the probability distributions and spectrogram representation. Concerning the transmission of the real-time video data over IP network, we use UDP (User Datagram Protocol) protocol via Wi-Fi connection.

The rest of this paper is organized as follows. A principle generation of the hyperchaotic encryption keys generation with C# implementation is given in Section 2. Section 3 further describes the principle of the proposed hyperchaos-based cryptosystem. Section 4 shows a graphic user interface of developed software application. Image and audio security analysis of the cryptosystem is given in Section 5 and Section 6 respectively. Conclusions are finally drawn in Section 7.

2. ROBUST HYYPERCHAOTIC ENCRYPTION KEYS

To generate robust encryption keys, we use the hyperchaotic Lorenz system which is described by the following nonlinear dynamic equations [19]:

$$\begin{cases} \dot{x} = a(y-x) \\ \dot{y} = x(b-z) - y + w \\ \dot{z} = xy - cz \\ \dot{w} = -dx \end{cases} \tag{1}$$

where x, y, z and w are four state variables, and a, b, c and d are positive real constants. The system is hyperchaotic for the parameters values a = 10, b = 28, c=8/3 and d =5, and with the initial conditions values x_0= -10, y_0 = -10, z_0= -10 and w_0 = -10.

Chaotic Encryption Keys Problem

It should be noted that the generated hyperchaotic keys encoded on 64-bits floating point data format suffer from long sequences of zeros and ones particularly in the integer parts. This leads to generate encryption keys with bad statistical performances, which is verified by testing them in the NIST test battery [20]. Thus, to overcome this problem and that of finite precision, we should increase the fractional data length. As a solution, we propose to increase the data length of the fractional part and decrease that of the integer part, and choose the fractional part as the encryption keys. This solution is the basic idea of the proposed Hyperchaos-based RNG (HRNG) for which the principle detail is given in the next subsection. This is the same idea used in [7] for fixed-point data format, it has given good results for hardware FPGA implementation of continuous chaotic systems.

HRNG Principle

The proposed HRNG principle scheme is illustrated in Fig. 1. It is composed by an hyperchaotic Lorenz generator implemented previously in C# using RK-4 method. After the generation of the four hyperchaotic signals x, y, z and w, we take only the fractional parts F(.) of the hyperchaotic samples. To obtain the hyperchaotic encryption keys K_1, K_2, K_3 and K_4, we encode on 64-bits double precision floating point data format, the obtained fractional parts $F(x)$, $F(y)$, $F(z)$ and $F(w)$. Note that the use of 64-bit double precision permits to overcome the problem of finite precision. Also it is possible to obtain long encryption keys multiple of 64 as 128, 192, 256 and so on by using concatenation operation. From this idea, we can express mathematically the HRNG equation system as fellow:

$$\begin{cases} K_1 = F(x) \\ K_2 = F(y) \\ K_3 = F(z) \\ K_4 = F(w) \end{cases} \tag{2}$$

In fig. 2, we present the analog representation of the encryption keys K_1, K_2, K_3 and K_4 and different 2D attractors (K_1/K_2, K_1/K_3, K_1/K_4) (Fig. 2(a)) compared to that of the hyperchaotic Lorenz system *x*, *y*, *z* and *w,* and 2D hyperchaotic attractors (x/y, x/z, x/w) respectively (Fig. 2(b)). Note that the obtained encryption keys K_1, K_2, K_3 and K_4 have the same behavior of that noise signals and the 2D attractors show that the phase spaces are completely and randomly occupied by the hyperchaotic trajectories.

To confirm and validate the claimed performance, we must evaluate the quality of randomness of the proposed HRNG. For this, statistical tests of the generated 64-bit encryption keys are commonly performed using the standard NIST SP 800-22 statistical test suite [20]. Table 1 summarizes the results of NIST test for the proposed HRNG. In this test suite, each test was performed 300 times on 1 Mbit substrings. A single test is considered as passed if the *P-value* is above the significance level of 0.01 or below 0.99 [20]. However, in Table 1, we show the measured values of *P-value*$_T$ knowing that if *P-value*$_T \geq 0.0001$, then the sequences can be considered to be uniformly distributed and the minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately 0.972766 for sample size equal to 300 binary sequences, for more details see the reference [20].

From the results of Table 1, we note that the encryption keys generated by the HRNG pass all kinds of the NIST test. Thus, we can say that our HRNG generates encryption keys with good statistical performances, i.e., good randomness keys.

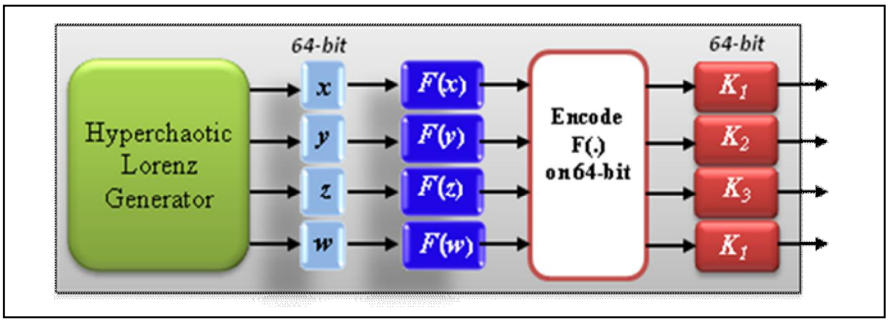
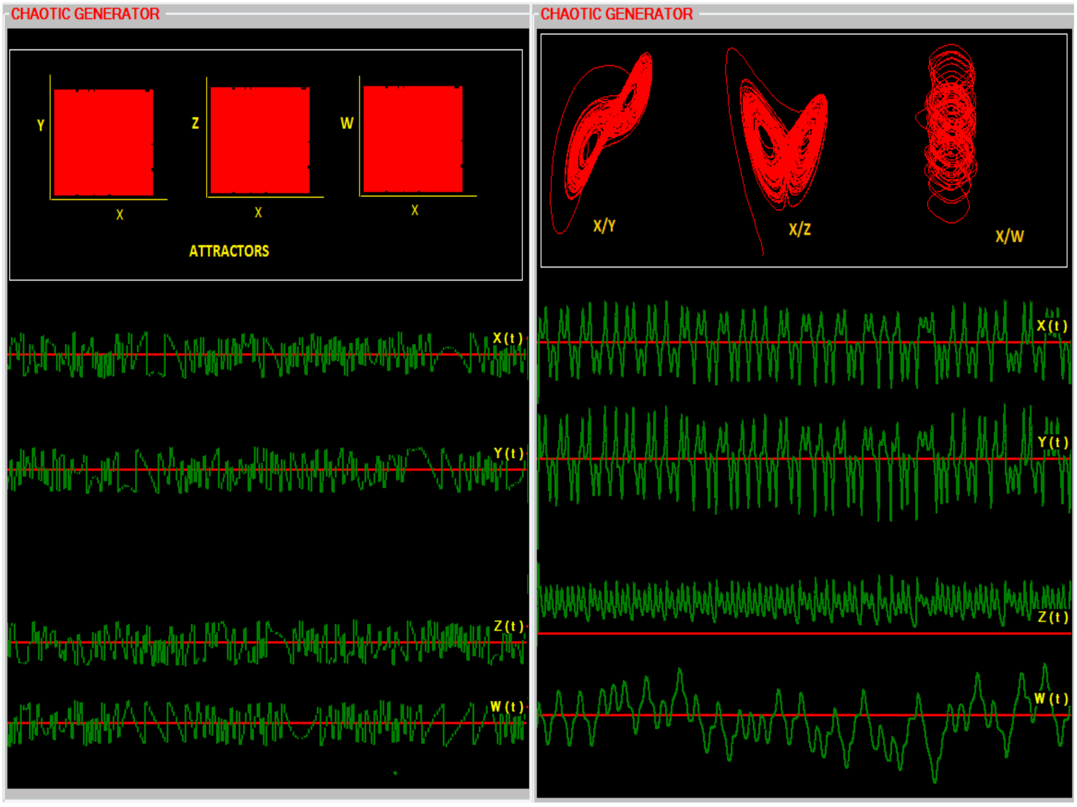


Fig. 1 HRNG principle scheme.



(a) (b)

Fig. 2 C# Simulation result: (a) The analog representation of the encryption keys K_1, K_2, K_3 and K_4 and (b) The hyperchaotic 2D attractors and signals (x, y, z, w).

Table 1 Nist Test Results

Statistical Tests	P-VALUE$_T$	Proportion
Frequency	0.644060	0.9800
Block-Frequency	0.023545	0.9833
Cumulative Sum up	0.671779	0.9900
Cumulative Sum down	0.699313	0.9933
Runs	0.117661	0.9933
Longest-run	0.630178	0.9967
Rank	0.840081	0.9867
FFT	0.100508	1.0000
N.P. templates	0.588652	0.9767
Over. Templates	0.547637	1.0000
Universal	0.568055	0.9833
App. Entropy	0.162606	0.9933
R.Excursions	0.249991	0.9947
R.E variant	0.711827	0.9840
Serial 1	0.706149	0.9833
Serial 2	0.110952	0.9833
Lempel ziv	0.071177	0.9900
L. Complexity	0.487885	0.9967

3. PROPOSED HYPERCHAOS-BASED CRYPTOSYSTEM

The principle scheme of the proposed hyperchaos-based multimedia cryptosystem is presented in figure (3). The encryption process consists of XORing the compressed video data with dynamical hyperchaotic keys K_1 and the compressed audio data with other dynamical keys K_2. Note that these dynamical keys K_1 and K_2 are generated simultaneously by the proposed HRNG (Fig. 1). We say dynamical keys because the HRNG generates a great number of keys without any repetition. Knowing that in symmetric cryptographical system, it is important to not repeat the same key because if an eavesdrop identifies the keystream, it could use this information to recover the message. For that reason, continuous chaotic systems offers great advantage for the encryption keys generation in cryptography. The obtained video (or audio) keystream is then packetized to be streamed over UDP. Also, the Decryption process consists of XORing the received buffered stream data (video or audio) by the same hyperchaotic keys K_1 or K_2 respectively followed by decompression level and then visualization process (listening through speaker).

Compressed Video Encryption

For the compressed video encryption process, we can define the keys K_1 as fellow:

$$K_1 = \left[K_{11}, K_{12}, K_{13}, \cdots, K_{1M}\right] \tag{3}$$

where M is the number of the generated keys K_1 for the video encryption, Knowing that each key K_{1i} (i=1,…, M) is encoded on 64-bit. Thus, in the proposed encryption process, M is fixed by the length of video data stream. For example, if the data stream length is 1024 bytes and the key length is 8-bit, then M=1024.

At the receiver, it should be noted that, it is necessary to use identical hyperchaotic keystream for the decryption process. This is guaranteed by robust synchronization of the received data with the locally generated hyperchaotic keys by using UDP package.

Compressed Audio Encryption

We use the same reasoning as above for the compressed audio encryption process. In fact, we can define the keys K_2 as fellow:

$$K_2 = \left[K_{21}, K_{22}, K_{23}, \cdots, K_{2N}\right] \tag{4}$$

where N is the number of the generated keys K_2 for the audio encryption.

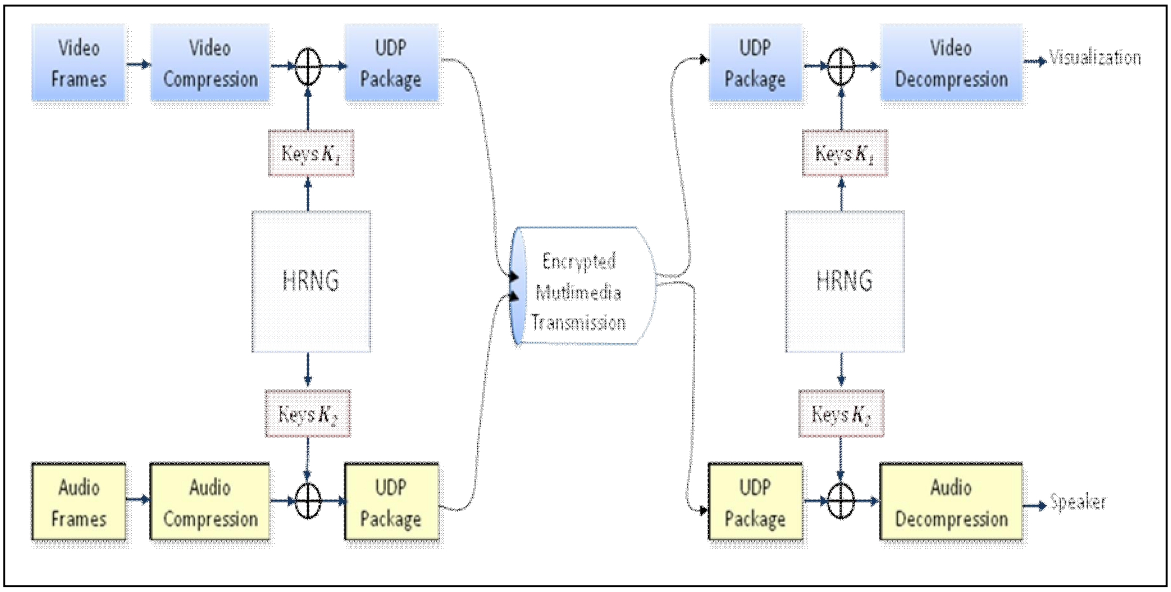


Fig. 3 Block diagram of the proposed hyperchaos-based multimedia cryptosystem.

4. APPLICATION

To validate the proposed hyperchaos-based multimedia cryptosystem, we have developed the software main form presented in the figure (4-a). This latter includes the necessary widgets for easier use. To evaluate performance for video delivery between two devices (laptops), we use Wi-Fi ad-hoc mode where the server device and the client device communicate directly. Experiments results for video and audio streaming with encryption and decryption process are represented in the figure (4-b) and figure (4-c) for server and client side respectively. The streaming video data from web camera and audio streaming data from microphone are played in real-time with a good quality and low latency.

5. IMAGE SECURITY ANALYSIS

In this section, we give the security analysis of the encryption image results. This latter must not be recovered by cryptanalyst if they want to break the cipher text. In this way, security analysis on chaos-based encryption is done by analyzing histogram cipher image, key sensitivity of the algorithms, statistical analysis using the correlation parameters and entropy and by looking at the key space available if carried out and analyze the cipher strength of a brute force attack [20].

Entropy

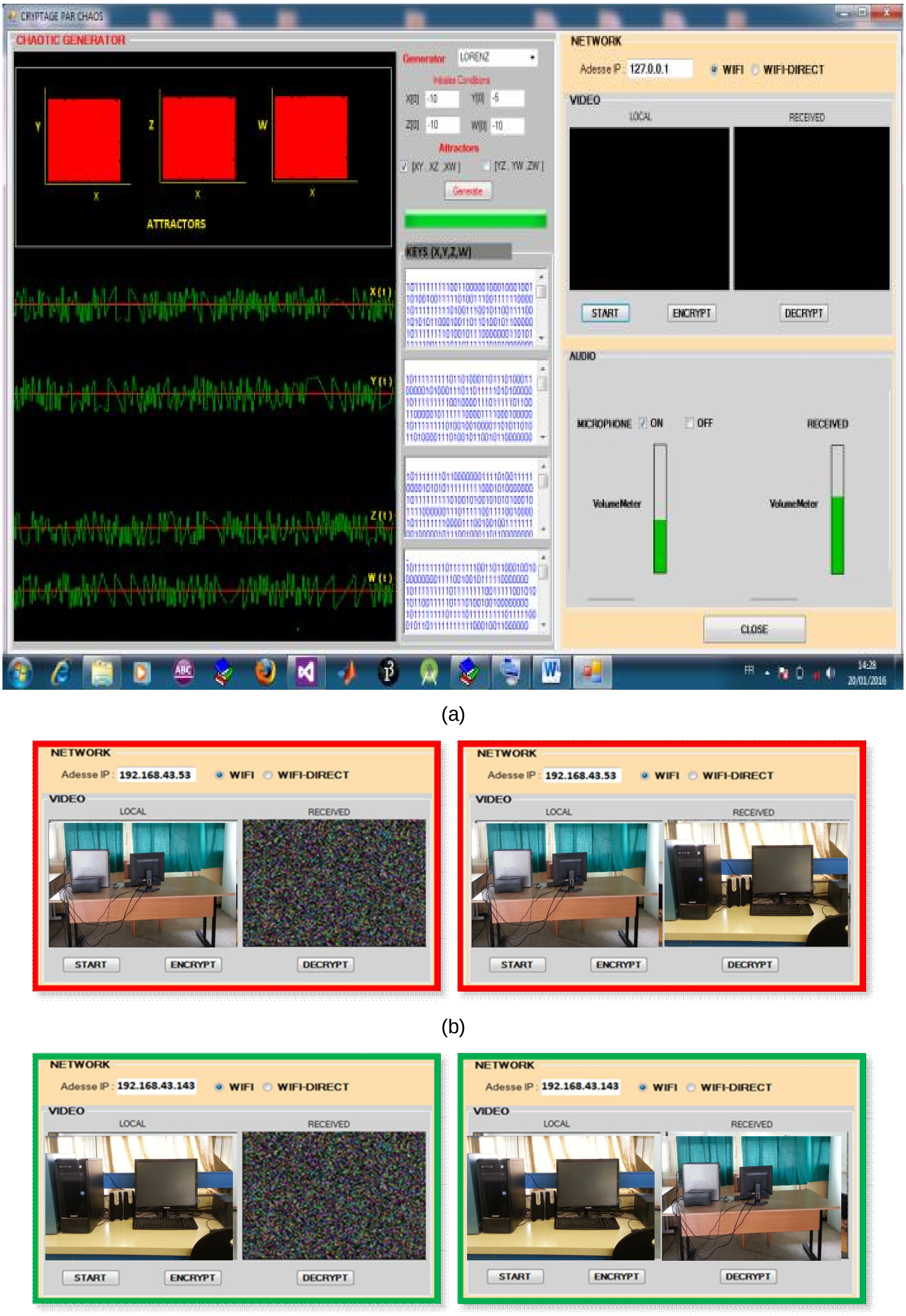
It is considered that ideal entropy value was 7.99902 (≈8). Thus, an encryption system designed safe from entropy attack, if it guaranty an entropy value ≈8 [20]. The result of the entropy calculation is shown in Table 2.

Table 2 Entropy Evaluation

Test Image	Size	Entropy Value
Mandaril	512 x 512	7.9993
Cameramman	256 x 256	7.9968

Image Correlation

It is known that an ideal encryption algorithm should produce the cipher images with no such correlation in the adjacent pixels (correlation ≈ 0) [20]. The result of the calculation of the correlation is shown in Table 3. These results indicate and confirm that the proposed cryptosystem possesses high security against statistical attacks.



Generator: LORENZ

Initial Conditions: X[0] -10 Y[0] -5 Z[0] -10 W[0] -10

Attractors: ☑ [XY XZ XW] □ [YZ YW ZW]

Adesse IP : 127.0.0.1 ● WIFI ○ WIFI-DIRECT

MICROPHONE ☑ ON □ OFF

(a)

Adesse IP : 192.168.43.53 ● WIFI ○ WIFI-DIRECT

Adesse IP : 192.168.43.53 ● WIFI ○ WIFI-DIRECT

(b)

Adesse IP : 192.168.43.143 ● WIFI ○ WIFI-DIRECT

Adesse IP : 192.168.43.143 ● WIFI ○ WIFI-DIRECT

(c)

Fig. 4 C# Main form of the application: (a) Server side, (left) received encrypted video from the client and (right) the decrypted video and (b) Client side, (left) received encrypted video from the server and (right) the decrypted video.

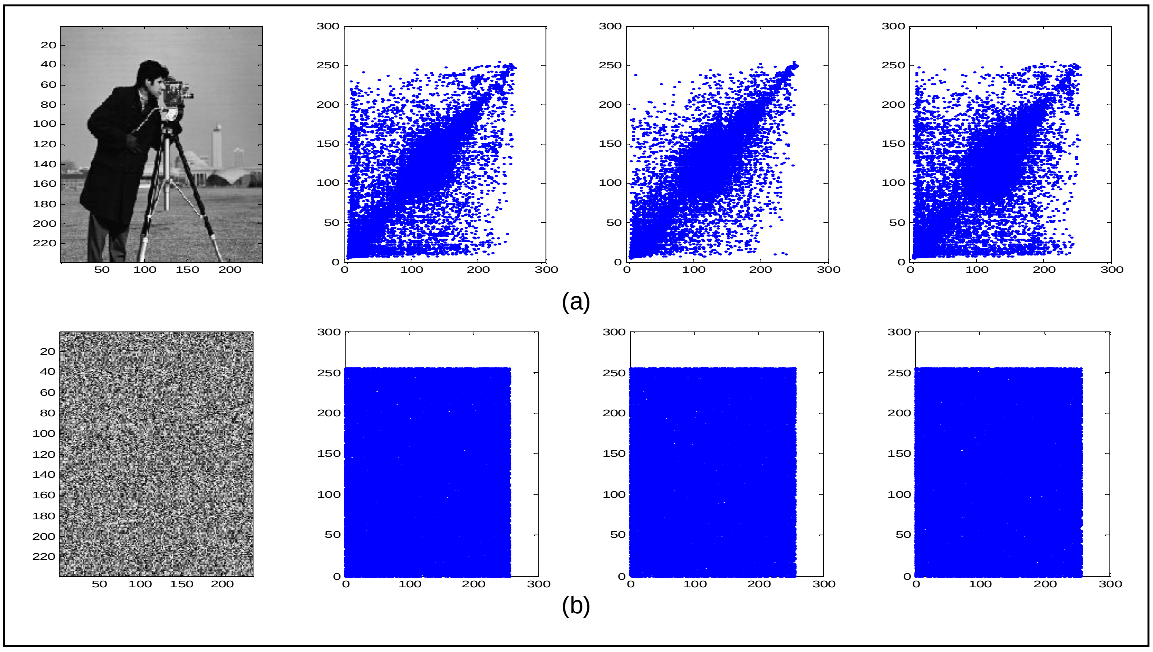


Fig. 5 Image correlation compute results: (a) Original Cameraman image and (b) Encrypted Cameraman image.

Table 3 Correlation Evaluation

Test Image	Size	Entropy Value
Mandaril	512 x 512	*0.*00128098
Cameramman	256 x 256	-*0.*00484255

Histogram of encrypted images

In cryptography, an ideal image encryption scheme should generate a ciphered image with different histogram from original images. As it is shown by the results in Fig. 9 and Fig. 10, the histograms of the ciphered images Mandril and Cameraman respectively are significantly different from original images, and allow no statistical resemblance to the plain image.

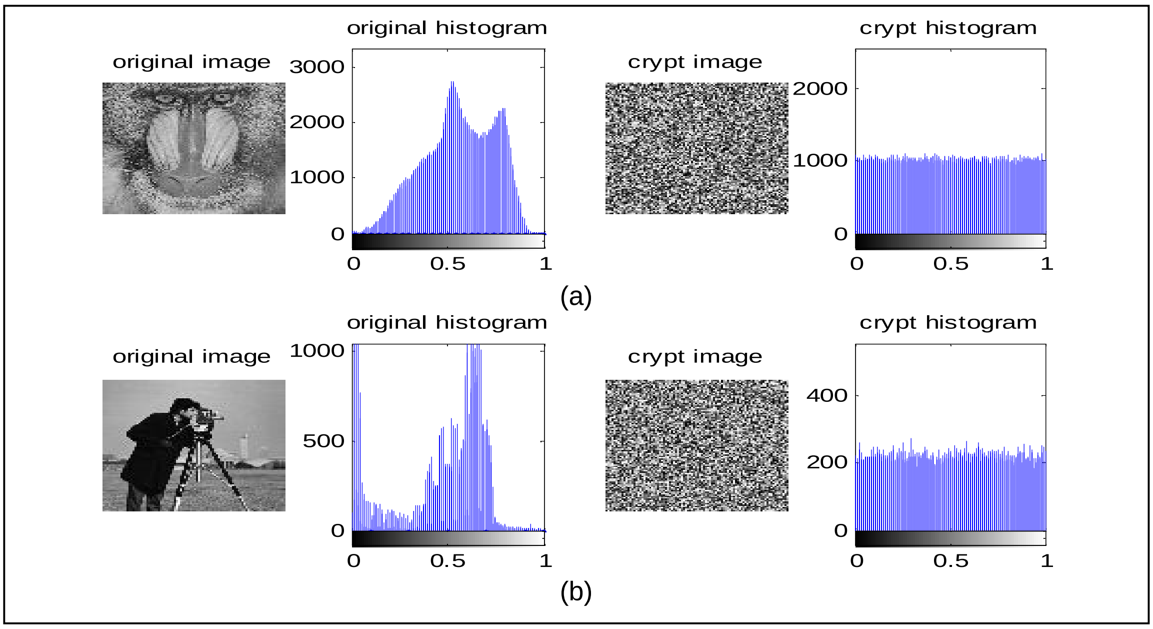


Fig. 6 Histogram of encrypted images: (a) Madril image and (b) Cameraman image.

Key Sensitivity

Another performance measure in cryptography is the guaranty of key sensitivity. This means that, the cryptosystem should be sensitive with respect to both the secret key and plain image. In fact, the change of a single bit in either the secret key or plain image should produce a completely different encrypted image [20].

The figures (11) and (12) give the results of decrypting the ciphered images Mandril and cameraman respectively with a difference of 10^{-14} between the two keys. These results show that the proposed cryptosystem is very sensitive to small difference in the secret key.

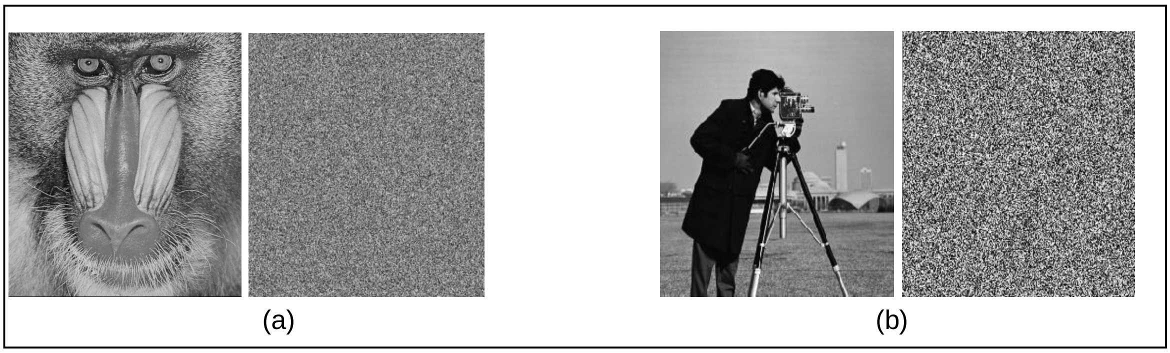


Fig. 7 Key sensitivity analysis: (a) madril image and (b) cameraman image.

Key Space

In image encryption domain, the secret key space should be large enough. This leads to make brute force attacks not possible [12]. For the proposed cryptosystem, the key space is estimated as follows:

- Secret Keys are the parameters (a, b, c, d) and the initial conditions (x_0, y_0, z_0, w_0) of the hyperchaotic Lorenz system. Thus, we have 8 parameter keys,
- Each secret keys is encoded on 64-bit,
- So it is likely that the key space is:

$$H(x_0, y_0, z_0, w_0, a, b, c, d)=2^{(8*64)}=2^{512}$$

We can conclude that the Key space of the proposed cryptosystem is large enough to resist the brute-force attack. In Table IV, we show a comparison of our key space value with that of [21] and [22].

Table 4 Evaluation of the Keys Space

Encryption scheme	Key Space
Ref. [21]	2^{266}
Ref. [22]	2^{349}
Proposed cryptosystem	2^{512}

6. AUDIO SECURITY ANALYSIS

In the figure (8), we present the audio security analysis including the autocorrelation computation, the probability distribution and the specteogram representation of the original and encrypted audio signals. In fig.(8-a), we present the original audio signal (left) and the encrypted signal (right). Figure (8-b) shows the autocorrealtions of the original signal (left) and the encrypted audio signal (right). The probability distributions, compred to that of normal dictribution, of the original signal (left) and the encrypted signal (right) are presented in figure (8-c). Finally, the figure (8-d) presents spectogram representation of the original (left) and encrypted signals (right). All of these results show and validate the robustness of the encryption audio process using the hyperchaotic encryption keys K_2.

7. CONCLUSION

In this paper, we have proposed hyperchaos-based cryptosystem for multimedia data security. To generate dynamic ciphering keys, an Hyperchaos-based Random Number Generator (HRNG) implemented on C# language is used. The basic idea is to exploit the four hyperchaotic signals of the Lorenz system to generate the ciphering keys with good statistical properties i.e, high quality of randomness. By passing all of the NIST tests, the proposed HRNG provides good quality random encryption keys which are used as keystream for video and audio encryption and decryption process via Wi-Fi network. In addition, the security analysis of the proposed hyperchaos-based cryptosystem has demonstrated and validated its robustness against several attacks. The proposed approach can be applied for securing all type of multimedia data transmission in divice to device communication systems.

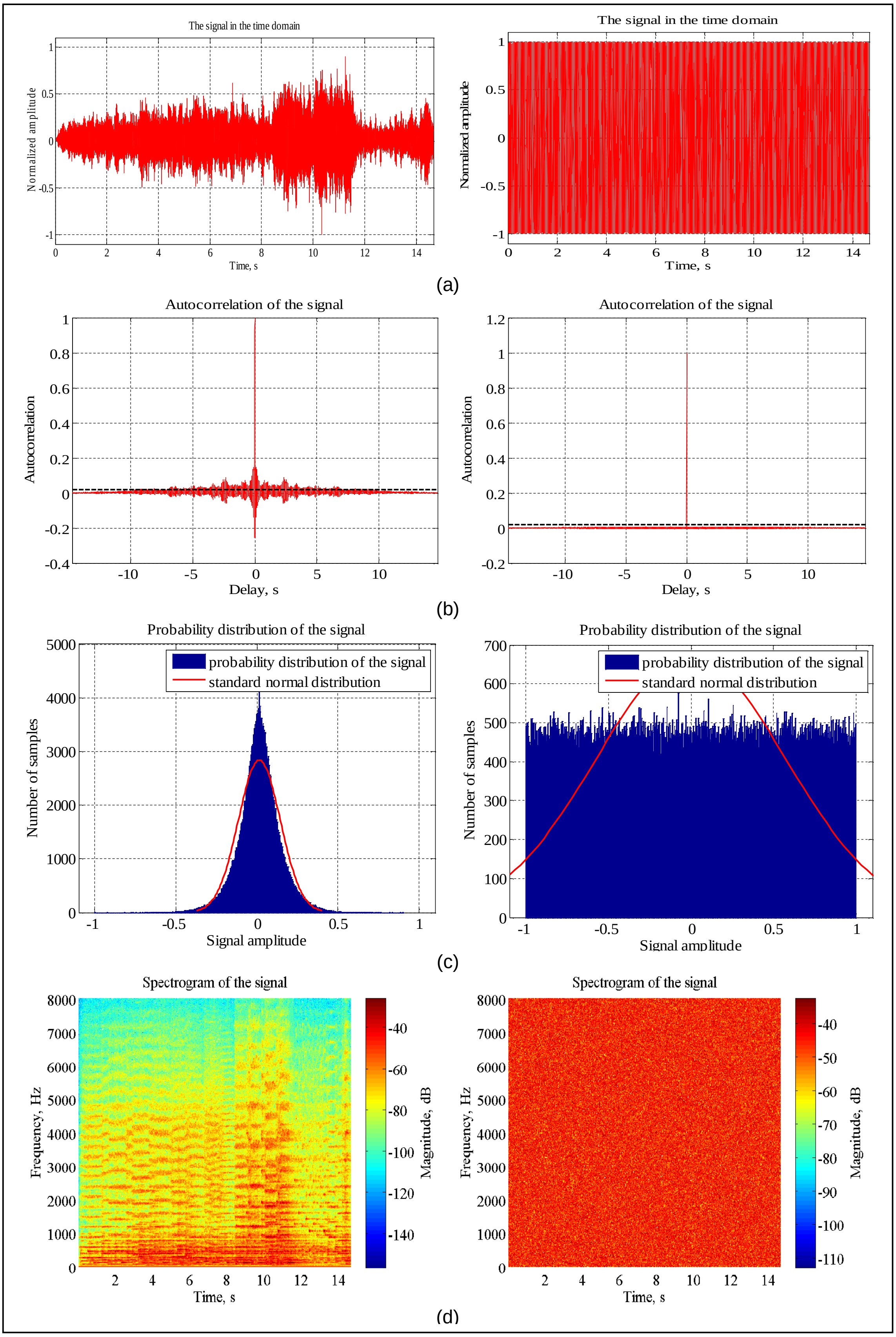


Fig. 8 Audio security analysis results: (a) Original audio signal (L) encrypted audio signal (R), (b) the autocorrelations of the original signal (L) and the encrypted audio signal (R), (c) the probability distributions of the original signal (L) and of the encrypted signal (R), (d) the spectrogram representation (L) and encrypted signals (R).

References

[26] S. Lian, “Multimedia Content Encryption: Techniques and Applications,” Auer-bach Publication, Taylor & Francis Group, 2008.

[27] W. Tuchman, “A brief history of the data encryption standard,” ACM Press, Addison-Wesley Publishing Co., New York, 1997.

[28] P. P. Dang and P.M. Chau, “Implementation IDEA algorithm for image encryption,” In Mathematics and Applications of Data/Image Coding, Compression, and Encryption III. Proceedings of SPIE, vol. 4122, 2000, pp. 1–9.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, “Introduction to algorithms,” 2nd edn. MIT Press, McGraw-Hill, Cambridge, 2001.

[30] FIPS Publication 197, the Advanced Encryption Standard (AES), U.S. DoC/NIST, November, 2001.

[31] Zhaopin Su, Guofu Zhang and Jianguo Jiang, “Multimedia Security: A Survey of Chaos-Based Encryption Technology,” Multimedia - A Multidisciplinary Approach to Complex Issues, Dr. Ioannis Karydis (Ed.), ISBN: 978-953-51-0216-8, 2012, InTech.

[32] S. Sadoudi, C. Tanougast and A. Dandache, “Hyperchaos-Based True Random Number Generator for Data Stream Encryption,” Book Chapter in “Progress in Data Encryption Research”, Nova Publisher, 2013.

[33] T. Yang, C.W. Wu, L.O. Chua, “Cryptography based on chaotic systems,” IEEE Trans. on Cirs and Sys.-I: Fundamental Theory and Applications, 44, 469–472 (1997).

[34] C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, “Image encryption process based on chaotic synchronization phenomena,” Signal Processing, 93, 2013, pp. 1328-1340.

[35] M. S. Azzaz, C. Tanougast, S. Sadoudi and A. Bouridane, “Synchronized hybrid chaotic generators: application to real-time wireless speech encryption,” Communications in Nonlinear Science and Numerical Simulation, 2013, 18(8): 2035-2047.

[36] S. Sadoudi, C. Tanougast, M. S. Azzaz and A. Dandache, “Design and FPGA implementation of wireless hyperchaotic communication system for secure real-time image transmission”, EURASIP Journal on Image and Vidéo Processing 2013, 2013:43.

[37] G. Alvarez, S. Li, “Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems”, Int. J. Bifurcat. Chaos Appl. Sci. Eng. 2006, 16, 2129–2151.

[38] G. Vidal, M.S. Baptista and H. Mancini, “A fast and light stream cipher for smartphones,” Eur. Phys. J. Special Topics, (2014).

[39] K. Ganesan, K. Murali, “ Image encryption using eight dimentional chaotic cat map”, Eur. Phys. J. Special Topics 223(8), 1611 (2014).

[40] P. Shanmugavadivu, P.S. Eliahim Jeevaraj, “ Adaptive integrated function systems filter for images highly corrupted with fixed-value impulse noise”, Eur. Phys. J. Special Topics 223(8), 1623 (2014).

[41] T.M. Hoang, D. Tran, “ Cryptanalysis and security improvement for selective image encrytption”, Eur. Phys. J. Special Topics 223(8), 1635 (2014).

[42] Y.B. Mao, G. Chen, S.G. Lian, “A novel fast image Encryption scheme based on the 3D CB Map□”, Int. J. Bifurcate Chaos, 2004, Vol. 14, pp. 3613–3624.

[43] Fengjian Wang, Yongping Zhang and Tianjie Cao, “Research of chaotic block cipher algorithm based on Logistic map□”, 2nd Int. Conference on Intelligent Computation Technology and Automation, 2009, pp. 678–681.

[44] R. Barboza, “Dynamics of a hyperchaotic Lorenz system,” Int. J. of Bifurcation and Chaos, vol. 17, no. 12, pp. 4285–4294, 2007.

[45] A. Rukhin1, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray , S. Vo, “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications,” NIST Special Publication 800-22 Revision 1a, April 2010.

[46] P. Irfan, Y. Prayudi and I. Riadi, “Image Encryption using Combination of Chaotic System and Rivers Shamir Adleman (RSA),” Int. J. of Computer Applications, Vol. 123, N.6, 0975–8887.

[47] M. A. B. Younes, A. Jantan, “Image Encryption Using Block-Based Transformation Algorithm,” IAENG, International Journal of Computer Science, 35:1, 200