A Hash-based Image Encryption Algorithm

Cheddad, A., Condell, J., Curran, K and Mc Kevitt, P
School of Computing and Intelligent Systems, Faculty of Computing and Engineering
University of Ulster, Northland Road, Derry, BT48 7JL

Abstract-There exist several algorithms that deal with text encryption. However there has been little research carried out to date on encrypting digital images or video files. This paper describes a novel way of encrypting digital images with password protection using 1D SHA-2 algorithm coupled with a compound forward transform. A spatial mask is generated from the frequency domain by taking advantage of the conjugate symmetry of the complex imagery part of the Fourier Transform. This mask is then XORed with the bit stream of the original image. Exclusive OR (XOR), a logical symmetric operation, that yields 0 if both binary pixels are zeros or if both are ones and 1 otherwise. This can be verified simply by modulus (pixel1, pixel2, 2). Finally, confusion is applied based on the displacement of the cipher's pixels in accordance with a reference mask. Both security and performance aspects of the proposed method are analyzed, which prove that the method is efficient and secure from a cryptographic point of view. One of the merits of such an algorithm is to force a continuous tone payload, a steganographic term, to map onto a balanced bits distribution sequence. This bit balance is needed in certain applications, such as steganography and watermarking, since it is likely to have a balanced perceptibility effect on the cover image when embedding.

I. INTRODUCTION

Much research has been done in the area of steganography which is the science of concealing data in a transmission medium in such a way that it would not draw the attention of eavesdroppers. Steganography has various useful applications such as for human rights organizations (since encryption is prohibited in some countries), smart IDs where individuals' details are embedded in their photographs (content authentication), data integrity by embedding checksum, medical imaging and secure transmission of medical data to name a few.

Various algorithms have been proposed to implement steganography in digital images. They can be categorized into three major clusters, algorithms using the spatial domain such as S-Tools [1], algorithms using the transform domain such as F5 [1] and algorithms taking an adaptive approach combined with one of the former two methods, e.g., ABCDE (A Block-based Complexity Data Embedding) [2]. Most of the existing steganographic methods rely on two factors: the secret key and the robustness of the algorithm. However, all of them either do not address the issue of encryption of the payload prior to embedding or merely give a hint of using one or more of the conventional block cipher algorithms. Hence, Westfeld et al. concluded their CRYSTAL (Cryptography and encoding in the context of steganographic algorithms) project with an important observation that "Crypto-Stego interaction is not very well researched yet" [3].

The renowned generic block cipher algorithms, such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA), etc, are not suitable to handle bulky data, i.e., digital images, due to their intensive computational process [4] [5] unless accelerated by hardware implementations. Additionally, such symmetric-key cryptographic algorithms are found unfit for digital images characterized with some intrinsic features such as bulk data capacity and high pixel correlation and redundancy

[6] [7], especially when confidentiality is needed. Other limitations were reported in [8] in light of multimedia communication.

Various hash algorithms are available such as MD5 (Message Digest 5) and SHA-2 (Secure Hash Algorithm) which hash data strings, thus changing their state from being natural to a seemingly unnatural state. A hash function is more formally defined as the mapping of bit strings of an arbitrary finite length to strings of fixed length [9]. Here the aim is to extend SHA-2 (the terminology and functions used as building blocks are described in the US Secure Hash Algorithm documentation) to encrypt digital 2D data. The introduction of two transforms combined with the output of SHA-2 creates a strong image encryption setting.

Security systems are built on increasingly strong cryptographic methods that foil pattern and statistical analysis attempts. Encryption is particularly useful for Intellectual Property Management and Protection (IPMP) standardization group and multimedia communications that prefer handling media streams compliant to certain multimedia coding standard, such as JPEG or MPEG-1/2/4 standard [10]. The proposed approach retains the structures readily available in the unencrypted bit stream. Such structures, often specified by special header patterns, would comply with standard multimedia codec. Thus, an encrypted video for instance would be still decoded successfully.

This paper is organized as follows; section II discusses related work, followed by the proposed encryption method in section III and security analysis is provided in section IV. Section V provides results and discussions. Finally, a conclusion is drawn in section VI.

II. Related Work analysis

The research on the design of secure image encryption tends to focus on transferring images into chaotic maps. Chaos theory, which essentially emerged from mathematics and physics, deals with the behaviour of certain nonlinear dynamic systems that exhibit a phenomenon under certain conditions known as chaos which adopt the Shannon requirement on diffusion and confusion [11]. Due to its attractive features such as its sensitivity to initial conditions and the random-like outspreading behaviour, chaotic maps are employed for various applications of data protection [9]. In the realm of 2D data, Shih [12] outlines the following method, called *Toral automorphism map*, in order to spread the neighbouring pixels into largely dispersed locations. The Transformation is represented through the following formula:

$\det\left(\begin{bmatrix} 1 & 1 \\ l & l+1 \end{bmatrix}\right) = 1 \quad or \; -1$, and l and N denote an arbitrary integer and the width of a square image respectively. Also, x and y represent the original pixel coordinates and x' and y' the new location coordinates. We refer to the determinant here as 'det'. Applying Eq. 1 to the sample image 'Lena', after exactly 17 iterations, termed as the *stable orbit*, the chaotic map converged into the original image. This Discrete Time Dynamic System (DTDS) is also the basic framework used in [13]. Regarding this method, it is important to note:

A) Since the algorithm uses a determinant in its process, the input matrix can only be square. This constraint was highlighted also in [4]. A work around this problem might be in applying the algorithm on square blocks of a given image repetitively. However, that would generate noticeable peculiar periodic square patterns given the nature of the process and of course this is not an interesting fact as it conflicts with the aim of generating chaotic maps.

B) As far as the security systems are concerned, the convergence of the translated pixels into their initial locations, i.e., image exact reconstruction after some iteration, is also not an appealing factor. This is an observed phenomenon in a variety of chaotic based algorithms.

Given one of the iterations is used, if an attacker gains knowledge of the algorithm and obtains the parameter "*l"*, which is actually not difficult to crack using a brute force attack, he/she will be able to invest some time to add more iterations that will reveal the original image. For example, Wang et al. [14] show that for such systems if two parameters are set to 10 and 8, then regardless of image contents, any image with the dimensions of 256x256 will converge after 128 iterations. This periodicity brings insecurity to the process as methods for computing the periodicity can be formulated such as the one proposed in [15] [16].

In a more detailed and concise attempt to introduce image encryption, Pisarchik et al. [17] demonstrated that any image can be represented as a lattice of pixels, each of which has a particular colour. The pixel colour is the combination of three components: red, green, and blue, each of which takes an integer value C= (Cr, Cg, and Cb) between 0 and 255. Thus, they create three parallel CMLs (Chaotic Map lattices) by converting each of these three colour components to the corresponding values of the map variable, $x_c = (x_c^r, x_c^g, x_c^b)$ and use these values as the initial conditions, $x_c = x_0$. Starting from different initial conditions, each chaotic map in the CMLs, after a small number of iterations, yields a different value from the initial conditions, and hence the image becomes indistinguishable because of an exponential divergence of chaotic trajectories [17]. They introduced seven steps for encrypting images and seven steps for decryption. Moreover, four parameters were used of which two were regulated. Their settings can have a tremendous effect on the chaotic map quality. Therefore, the receiver must know the decryption algorithm and the parameters which act as secret keys.

The algorithm is well formulated and adequately presented; it yields good results for RGB images as proclaimed by the authors. It was noticed that they used a rounding operator which was applied recursively along the different iterations. The major concern would be in recovering the exact intensity values of the input image as the recovered image shown in their work might be just an approximation because of the aforementioned operator. This is important, especially in the application of steganography where the objective is to recover the exact embedded file rather than its approximation. The raised point was remarked independently in [18] where they stated that a sensitive generator, i.e., a generator with a rounding operator, can produce two different binary sequences (after some iterations) for the same initial values and parameters if generated on two different machines which round off fractions after unmatched decimal places. However, a desired algorithm must be efficient, repeatable and portable (i.e., works in the same way in different software/hardware environments) [19]. As a result, such a chaotic encryption system is not invertible under double precision arithmetic [20].

Usman et al. [4] describe a method for generating chaotic maps to encrypt medical images by repetitive pixel arrangement and column and row permutations. The pixel arrangement is achieved through the following system:

$$\begin{aligned} &X(i,j) \rightarrow Y(k,l), where \\ &k = \lfloor (j + (i-1)N - 1)/L \rfloor + 1 \\ &l = (j + (i-1)N - 1) \bmod (L) + 1 \end{aligned} \qquad (2)$$

Here, *k, l* denote the mapped spatial coordinates of the original location at *i, j*. *N* and *L* are the height of the original image and transformed image respectively in such a way that: $(KxL) = (MxN), where: K \neq M$. The authors show some experiments in which the deciphered phase was missing. It is suspected that the rounding operator introduced in Eq. 2 will force some pixels to collude at the same location resulting in the loss of information needed for the original image reconstruction. Zou et al. [21] reduce the number of iterations using 2D generalised Baker transformations to enhance the key space.

Unfortunately, most chaotic maps are unstable due to the periodicity of the mapping [13] [22]. Systems based on such maps are prone to attacks, such as the broken system shown in [23].

Other types of image encryption include the Fourier plane encoding algorithm, introduced in [24], which is attacked in [25] using an initial guess of the Fourier plane random phase while searching over a key space to minimise a cost function between the decrypted image for a given key and the original image. This spurred a variety of authors to apply the Fourier transform such as the works in [26] [27].

Shin and Kim [28] presented a phase-only encryption scheme using the Fourier plane. To generate this phase encrypted data, a zero-padded original image, multiplied by a random phase image, is Fourier transformed and its real-valued data is encrypted with key data by using phase-encoded XOR rules. Since the original information is encrypted on the Fourier plane, the decryption cannot retrieve the original image without perceptual degradation, i.e., PSNR in the interval (20dB, 42.23dB).

One time pad hash algorithms, known also as stream ciphers, were believed to be unsuitable for image encryption since they would require a key of the size of the ciphered image itself [4]. Sinha and Singh [29] use MD5 to generate image signature by which they encrypt the image itself using a bitwise exclusive-OR (XOR) operation; they coupled that with an error control code, i.e., Bose-Chaudhuri Hochquenghem (BCH). The ciphered image was larger than the original because of the added redundancy due to applying the BCH. Since the message digest is smaller than the image, they XOR the signature block by block which eventually left some traces of repetitive patterns. Hence, their method was commented on in [30] in which they show also how insecure the method is with some experiments, a fact that provoked Sinha and Singh [29] to debate the arguments in their recent published reply in [31].

Gao and Chen [32] propose an image encryption algorithm based on hyper-chaos, which uses a matrix permutation to shuffle the pixel positions of the plain-image, i.e., logistic map, and then the states combination of hyper-chaos is used to change the grey values of the shuffled-image (diffusion). Their proposed algorithm did not survive attacks for too long. Rhouma and Belghith [33] successfully broke such cryptosystem using a chosen plaintext attack and a chosen cipher-text attack that recover the ciphered-image without any knowledge of the key value.

Zeghid et al. [5] propose a new modified version of AES which involves the design of a secure symmetric image encryption technique. The AES is extended to support a key stream generator for image encryption which can overcome the problem of textured zones existing in other known encryption algorithms. The problems of AES based algorithm are, computational time complexity, generation of repetitive spatial patterns, and the sensitivity to image manipulation.

In the realm of information hiding some steganographic applications prefer to use conventional pseudo-random number generator (PRNG) algorithms which form the basic and essential ingredient for any stochastic simulation in which random variables and other random objects are simulated by deterministic algorithms [19].

III. Proposed Encryption Method

This proposal exploits the strength of a 1D hash algorithm, namely SHA-2 and extends it to handle 2D data such as images. “Secure one-way hash functions are recurring tools in cryptosystems just like the symmetric block ciphers. They are highly flexible primitives that can be used to obtain privacy, integrity and authenticity” [34].

SHA-2 hash standard underlies four secure hash algorithms SHA-224, SHA-256, SHA-384, and SHA-512. These algorithms are the result of a continuous development of SHA-1. SHA algorithms are used to provide a condensed fixed length representation known as message digest of an input message. The length of the unique digest ranges from 160 to 512 bits depending on the used algorithm [35]. The security of SHA-256, SHA-384, and SHA-512 matches the security of AES with complexity of the best attack as 2^{128}, 2^{192} and 2^{256}, respectively, have been announced by the National Institute of Standards and Technology (NIST) [36].

SHA-2 can be described in two phases: pre-processing and hash computation. Pre-processing comprises padding, parsing the padded message into m-bit blocks, and setting any initialization values to be used in the hash generation. The hash computation generates a message schedule from the padded message which is used, along with functions, constants and word operations, to iteratively generate a series of hash values [37]. For exhaustive details on SHA-2 algorithms see the online specification[1].

The DCT and FFT are incorporated into the process to increase the disguise level and thus generate a random-like output that does not leave any distinguishable patterns of the original image. The ordering of the transforms is very crucial since the algorithm's strength lands itself to exploiting the symmetrical property of the FFT's imaginary part. The exhaustive step by step description of the encryption algorithm is illustrated in Fig. 1. The method works as a one-time pad cipher; therefore, the decryption will follow the same digital process but with the cipher input into the system, i.e., symmetric encryption. Starting with a password phrase $\boldsymbol{K}$ supplied by the user the algorithm generates a SHA-2, i.e., SHA-256, based hash string $\boldsymbol{H(K)}$ which forms the initial condition. The vector $\boldsymbol{H}$, treated as a string of hexadecimal characters, is then converted to its decimal version and finally transformed to a bit stream matrix of fixed dimension $\boldsymbol{K}'=\{8x32\}$. Parallel to this, the original image $\boldsymbol{A}$ is converted to a bit stream and reshaped to the order $8xMN$.

The partially extended key, herein $\boldsymbol{K}'$, is still short to accommodate the image bit stream. Therefore, the algorithm performs key full expansion towards the needed dimension, herein $8xMN$. Obviously, this step would result in repetitive patterns that would make the ciphered image prone to attacks, a problem that was independently noticed in [4]. To cope with this situation the method applies a thresholded *DCT*, where Eq.4 is used, followed by *FFT* to provide the diffusion requirement and to tighten the security. Note that nested transforms are not scant in the literature, for example O'Ruanaidh and Pun [38] used *FFT* followed by log-polar mapping and *FFT* to embed a watermark.

Let the resized key be $\lambda_{8,MN}$ where the subscripts *M* and *N* denote the width and height dimensions of the image, respectively. The FFT operates on the DCT transform of $\lambda_{8,MN}$ subject to Eq. (4):

$$f(u,v)=\frac{1}{8MN}\sum_{x=0}^{7}\sum_{y=0}^{MN-1}F(x,y)e^{-2\pi i(xu/8+yv/MN)}$$

$$,satisfying \quad Eq.\ (4) \qquad (3)$$

$$where, F(x,y)=DCT(\lambda_{8,MN}), subject \quad to \quad (*)$$

$$F(x,y)=\begin{cases}1 & if \quad DCT(\lambda_{8,MN})>0\\ 0 & Otherwise\end{cases} \qquad (*)$$

$F(x, y)$ is the thresholded DCT of the resized 2D bit stream of the 1D hash string generated from applying SHA-2 and denoted herein by $\lambda_{8,MN}$. Similarly $f(u,v)$ is the thresholded Fourier Transform of the function $F(x,y)$.

[1] US Secure Hash Algorithms, (2006). Secure Hash Standard [Online]. Available: http://www.ietf.org/rfc/rfc4634.txt.

Generating a pseudo-random binary sequence from the orbit of $f(u,v)$ requires the mapping of the state of the system to its binary values $\{0,1\}$. One clear method for converting a real number to a discrete bit symbol is to use a rule as shown in Eq. 4. Given the output of Eq. (3) we can derive the corresponding binary map as depicted in Fig. 2:

$$Map(x,y)=\begin{cases}1 & if \quad imag(f(u,v))>thr\\ 0 & Otherwise\end{cases} \qquad (4)$$

where thr is an appropriately selected threshold value and $imag(\bullet)$ denotes the imaginary part of the complex function which can be compared directly with a threshold thr. For a balanced binary sequence and for robustness, thr should be chosen such that the probability $P(imag(f(u,v))<thr) = P(imag(f(u,v))>thr)$. Fortunately, the imaginary part of the signal $f(u,v)$ is always symmetrical around zero (see Fig. 6 for validity of this property). Therefore, $thr=0$ is an explicit solution. Since the coefficients in this calculation are converted to a binary map the reverse construction of the password phrase is impossible. Hence the name *Irreversible Fast Fourier Transform* (*IrFFT*). The generated bit-pattern exhibits sufficient randomness, which it will be proved, to provide cryptographic security as shown in the security analysis section. This map finally is *XORed* with the bit stream version of the image. The result is then converted into grayscale values and then reshaped to form the ciphered image. A post-processing step is introduced providing pixel substitution based on a new randomized image using K2=H(H(K)), this is illustrated with a simple example in Fig. 1 (bottom).

The coding phase uses the *Map* (Eq. 4) to encrypt the bit stream of image *A* and produce a new encrypted matrix A', in such a way that:

$$\varepsilon_{auth} \equiv \{A - D(A', Map)\} \qquad (5)$$

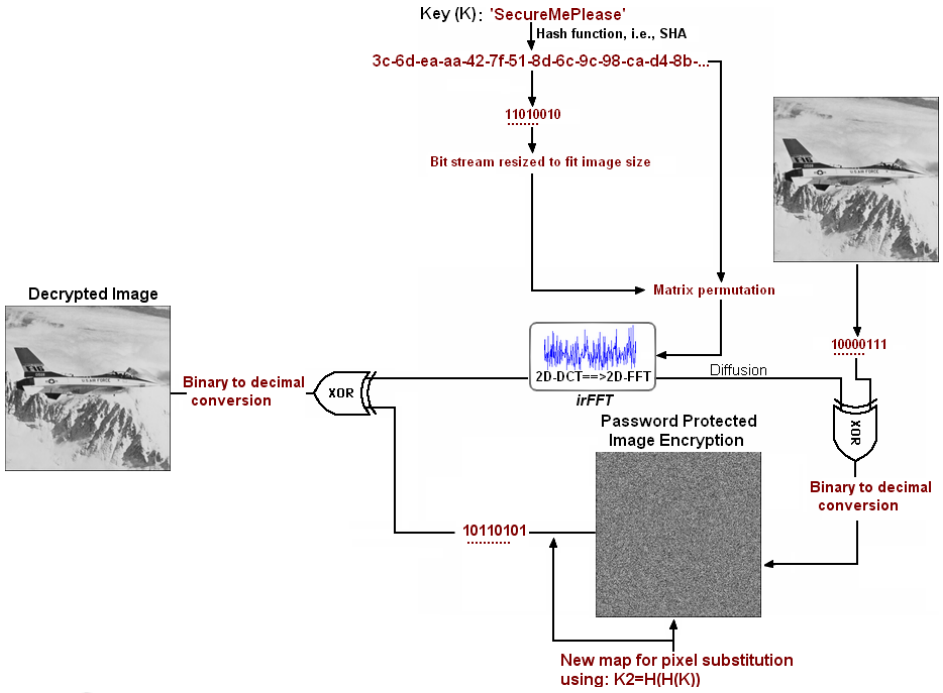
where, $D(A', Map)$ denotes the decoding of A' with the same key generated *Map*. Ideally, ε_{auth} should be equal to {Ø} and starts to deviate from that when A' undergoes an image processing attack.

Another noticed phenomenon which has been exploited was the sensitivity of the spread of the FFT coefficients to changes in the spatial domain. Therefore when we coupled this with the sensitivity of the SHA-2 algorithm to changes of the initial condition, i.e., password phrase, we can meet easily the Shannon law requirements. For instance slight changes in the password phrase will, with overwhelming probability, result in a completely different hash and therefore completely different *Map*.

In reference to Fig.1 the encryption procedure is summarised as follows:

1- Generate a SHA-based hash string of the user password.
2- Convert the hash into {8x32} binary matrix, SHA-256 is used. This step forms the initial key expansion.
3- Resize this matrix to fit into the image binary matrix, this is denotes as $\lambda_{8,MN}$
   where *M*, *N* are the original image dimensions.
4- Feed the result into a compound transform, DCT followed by FFT with those constraints described in Eq. 3&4.
5- XOR the result with the image bit stream
6- Generate a new randomized binary Map of size (*M* x *N*) for the post-processing step (pixel substitution) as shown with a simple example in Fig.1 (bottom)
7- Encrypted image.

The decryption process starts with step 6 as a pre-processing step then followed by steps 1-5.




```
>> Or

Or_before_Swap =

  193   125   120    19
  102   233   139     9
  254    12    27   124
  215   120   115   120

>> After swap using the new map for pixel substitution (map)
   Values falling on map(0) are replaced with values falling on map(1) and vice versa
Or_after_Swap =

  102   233    12   120
  193   125     9   139
  215   120   120   115
  254    27   124    19

>> map

map =

    0     1     1     0
    1     0     0     1
    1     0     1     1
    0     0     0     1
```

Fig. 1. Block diagram of the steps used in the proposed algorithm for image encryption and decryption. (bottom) shows a simple example to illustrate the swapping process used for pixel substitution.

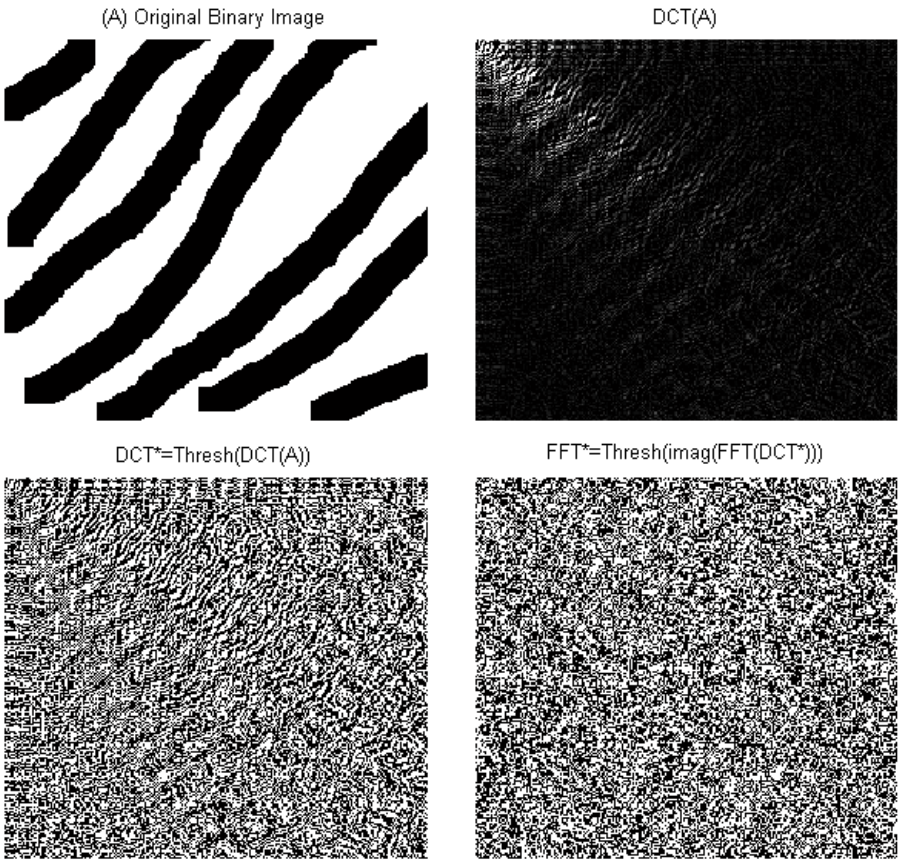


Fig. 2. Generation of random pattern using the frequency transform of the signal. Note that the pattern has the advantage of straightforwardly providing a balanced bit stream

So, the core idea here is to transform this sensitivity into the spatial domain where 2D-DCT and 2D-FFT can be applied so that to introduce the aforementioned sensitivity to the two dimensional space. As such, images can be easily encoded securely with password protection. Note that this scheme efficiently encrypts grayscale and binary images. However, for RGB images we noticed that using the same password for the three primaries will yield some traceable patterns inherited from the original image (RGB colours are highly correlated). This is easily overcome through the following two choices: either the user supplies three passwords each of which encrypts one colour channel or more conveniently the system generates other two unique keys from the original supplied password. For instance, a single key can be utilized to generate the following different hash functions $H(\vec{K}), H(\overleftarrow{K}),$ and $H(H(\vec{K}))$ to encrypt the R, G and B channels, respectively. K denotes the supplied key, the arrows indicate the string reading directions and $H(H(\bullet))$ denotes double hashing.

IV. SECURITY ANALYSIS OF THE ENCRYPTION METHOD

This section analyses the security aspects of the proposed method. Encryption algorithms are assumed to be robust to different statistical and visual attacks, moreover key sensitivity and key space should be adequate. In addition to that, and being a tailored method for

steganography, the result should exhibit high randomness and a balanced bit values. Therefore, this section is split into five sub-sections, namely, key space analysis, key sensitivity, adjacent pixels analysis, randomness and other security merits.

A. Key Space Analysis

The key space analysis of the proposed algorithm essentially involves analysing the underlying SHA-2 algorithm. SHA-2 accepts any key of any length less than 2^{64} bits. SHA is secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. SHA has been extensively adopted in several organisations and has received much scrutiny from the cryptography community. The proposed encryption algorithm is flexible enough to migrate to a newer version of the SHA's algorithmic family or other secure hash functions especially knowing that no collisions have been found in SHA-2. Any version of SHA's family, be it SHA-224, SHA-256, SHA-384, or SHA-512, can all be written as an {8x?} matrix where "?" denotes (the length of the hash/2) and 8 comes from the binary of the hexadecimal value of the ASCII pairs.

B. Key Sensitivity Analysis

As it was the aim from the design, the algorithm is proven to be very sensitive to initial conditions, see Fig. 3. That was due to the plugged in hash algorithm and the *FFT* which made the technique immune to malleability attack. Fig.3 (a) shows the encrypted "boat" image using the word Steganography as a password, note that this password is case sensitive, Fig.3(b) successful decryption of Fig.3(a) and Fig.3 (c, d) illustrate decryption failure with slightly different password and hash code shown in bold, respectively.

(a) (b) (c) (d)

Fig.3. Key sensitivity test: (a) encrypted image, (b) decrypted (a) using the key 'Steganography' '40662a5f1e7349123c4012d827be8688d9fe013**b**', (c) decrypted (a) using the wrong key 'Steganographie' 'c703bbc5b91736d8daa72fd5d620536d0dfbfe01' (d) decrypted (a) using slightly modified hash '40662a5f1e7349123c4012d827be8688d9fe013**B**'.

C. Adjacent Pixels Analysis

To test for statistical properties of the original image and the encrypted version we carried out tests based on the linear relationship between two adjacent pixels horizontally, vertically and diagonally. It is observed that natural images with natural data have a high correlation ratio between neighbouring pixels, see Fig.4. To measure this relationship the correlation coefficient was calculated of each pair of pixels (as shown in Table 1). The comparison given in Table 1 shows that the proposed method outperforms other recent methods reported in the literature. To establish a fair evaluation the same test image was used. In the horizontal, diagonal and vertical directions the encrypted version under this scheme had the highest performance. Bear in mind that unlike various methods, the proposed algorithm does not involve an extensive and computationally intensive iterations process. The encrypted image shown in Fig. 3 is automatically generated once the program is invoked with a key. The

process does not retain any image statistics. This can be seen by comparing histograms of the plain and encrypted images, the original histogram is flattened and has a uniform distribution.

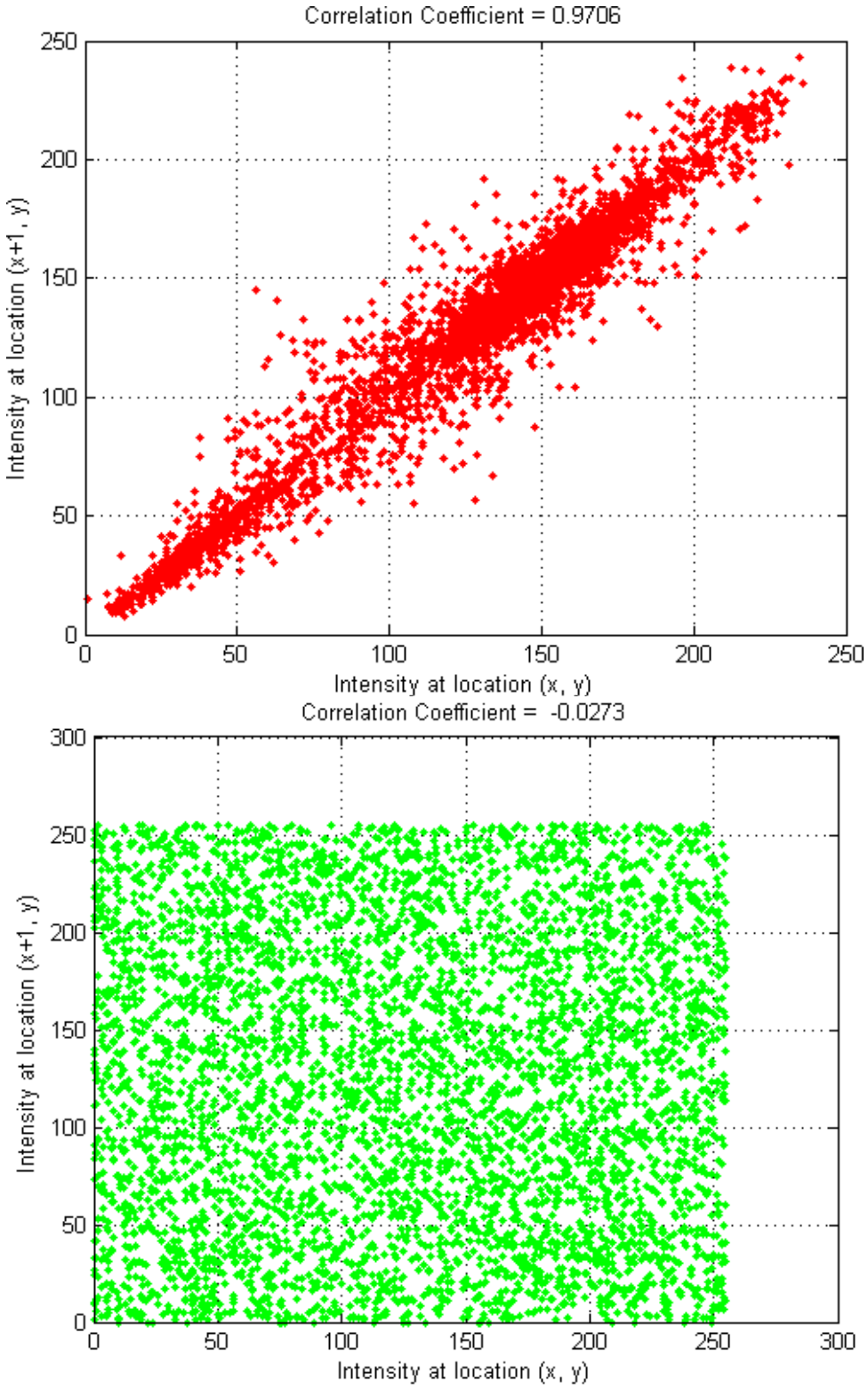


Fig. 4. Correlation analysis of 5000 pairs of horizontal adjacent pixels chosen randomly from: (top) the original plain boat image (Fig. 3 (b)), (bottom) the encrypted image (Fig. 3 (a)) using the proposed method.

D. Randomness test

In this test, the method is submitted to batteries of empirical tests to measure the quality of the generated random sequence. "It is impossible to give a mathematical proof that a generator is indeed a random bit generator" [39]. Bear in mind that there are countless number of possible statistical tests, each of which reports the presence or absence of a "pattern" which, if detected, would indicate that the sequence is nonrandom [40]. This section highlights some tests adopted from the statistical test suite published by the *National Institute of Standards and Technology* in August 2008.

Table. 1. Performance analysis of the proposed method with recent methods using Lena image - Correlation coefficients, ranging from '1' highly correlated to '-1' highly uncorrelated, of pairs of adjacent pixels in different directions. These coefficients ensure the two considered images are statistically independent.

Scan Direction	**Horizontal**	**Vertical**	**Diagonal**
Original Image	0.9194	0.9576	0.9016
Proposed	-0.0028	-0.0068	0.0044
PRNG	0.002291	0.005702	0.007064
[41]	0.002933	-0.004052	0.001368
[42]	0.006816	0.007827	0.003233
[43]	0.005343	0.008460	0.003557
[21]	0.01183	0.00872	0.01527
[5]	0.02	0.03	Not reported
[44]	0.0085	0.0054	0.0242
[45]	0.0171	0.0098	0.0330
[46]	0.01183	0.00016	0.01480

The Chi-square distribution is a very powerful statistical test. Its distribution can be used to compare the goodness-of-fit of the observed frequencies of events to their expected frequencies under a hypothesized distribution [39]. Fig. 5 shows clearly that the proposed cipher passes this test. The result is of no surprise knowing that the random bits were derived from a Gaussian distribution as shown in Fig. 6 (top).

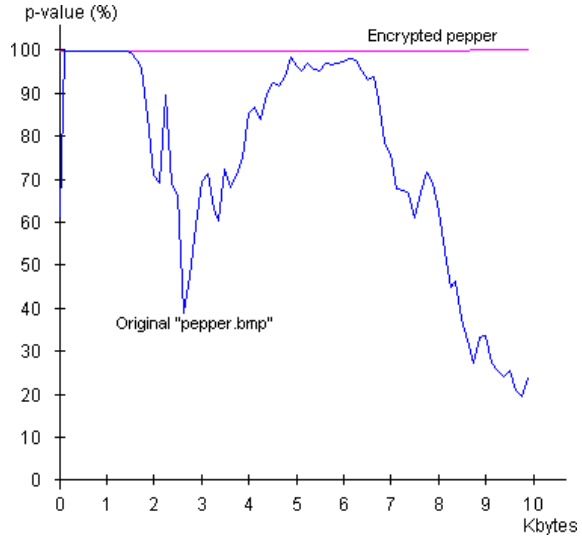


Fig. 5. The Chi-square χ^2 distribution of the original and encrypted signals.

Frequency test (monobit test): Given a randomly generated N-bit sequence, we would expect approximately half the bits in the sequence to map to ones and approximately half to map to zeros. The frequency test checks that the number of ones in the sequence is not significantly different from N/2 [18]. It is noticed that the complex imaginary part of the *Fast*

Fourier Transform exhibits conjugate symmetry. Fig. 6 exemplifies such a property where the magnitude of the transform is centred on the origin *imag* (*f (u, v)) = 0*. In other words, Eq. 4 yields a balanced binary sequence which passes this test. This assertion holds true for any 8-bit image as well as binary images.

Let the length of the encrypted bit string be *n* and let the generated bit sequence be given as $\varepsilon = \varepsilon_1, \varepsilon_2, ..., \varepsilon_n . where : \varepsilon_i \in \{0,1\}$. This sequence is summed up in the following manner: $S_n = X_1 + X_2 + ... + X_n, where : X_i = 2\varepsilon_i - 1 = \pm 1$. Now, the *P-value* can be computed using the *complementary error function* (*erfc*) as shown in Eq. 6.

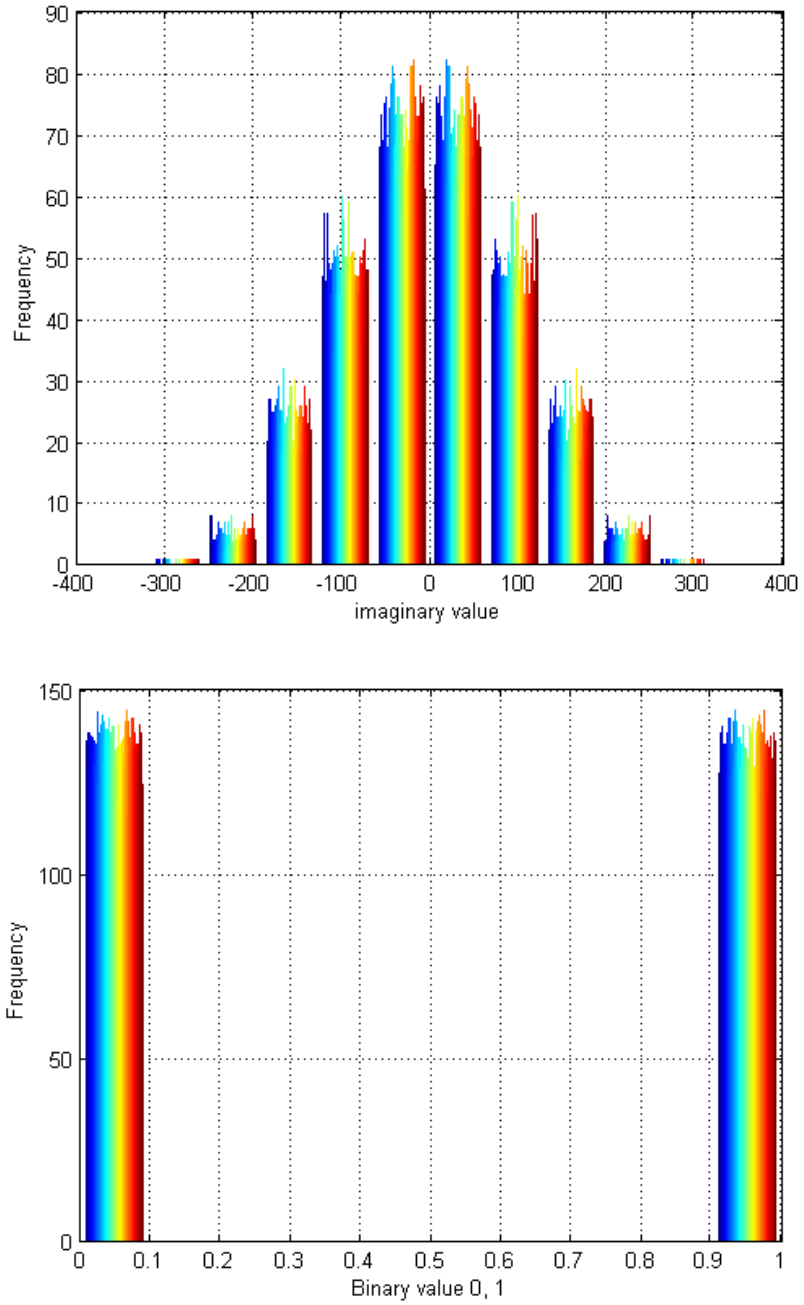


Fig. 6. Overcoming the frequency test, analysis: (top) the imaginary part of *f(u, v)* in Eq. 3, asserting that $P(\text{imag}(f(u,v)) > x) = P(\text{imag}(f(u,v)) < -x)$, for any *x* value (bottom) the corresponding binary map after applying Eq. 4, the number of non-zero matrix elements is 32766~=N/2, where N= 256x256=65536.

$$P-value = erfc\left(\frac{|S_n|}{\sqrt{2n}}\right) \quad (6)$$

Testing this on image "*pepper_encrypted.bmp*" yields:
P-value=erfc(0.3950/sqrt(2))= 0.6928. Since the *P-value* is ≥ 0.01 (decision rule at the 1% level, common values of α in cryptography are about 0.01 [40]) then we accept the bit sequence as random.

Another test was conducted on the image shown in Fig. 7 (left) demonstrating different smooth blocks. As can be seen, the proposed algorithm performs better than AES in confusing the structure of the image content and also in generating the needed balanced bit stream, see, Table 2 and Fig.8. This definitely serves justifying the final remarks we make about the unsuitability of AES algorithms in encrypting digital images.

Table 2. Monobit test, of our method against AES, used to construct Fig. 8

Bit plan\method	Proposed		AES	
	PNZ_n (*)	PZ_n	$AESNZ_n$	$AESZ_n$
1st	524741	523835	519587	528989
2nd	524678	523898	516426	532150
3rd	524061	524515	523456	525120
4th	524968	523608	500456	548120
5th	523821	524755	534373	514203
6th	523118	525458	485999	562577
7th	523248	525328	497225	551351
8th	524838	523738	555971	492605

* Z_n : Number of zeros and NZ_n: Number of non-zeros, i.e., 1s

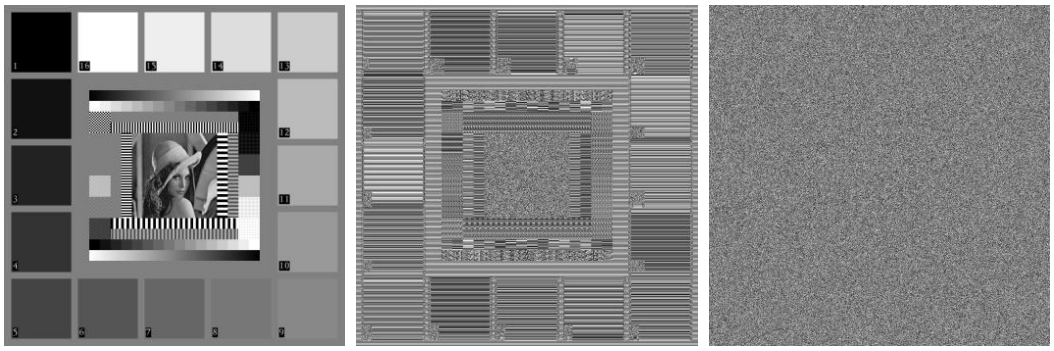


Fig.7. Performance of the proposed method against AES in confusing image structure: (left to right) original, encrypted using AES and using proposed method, respectively

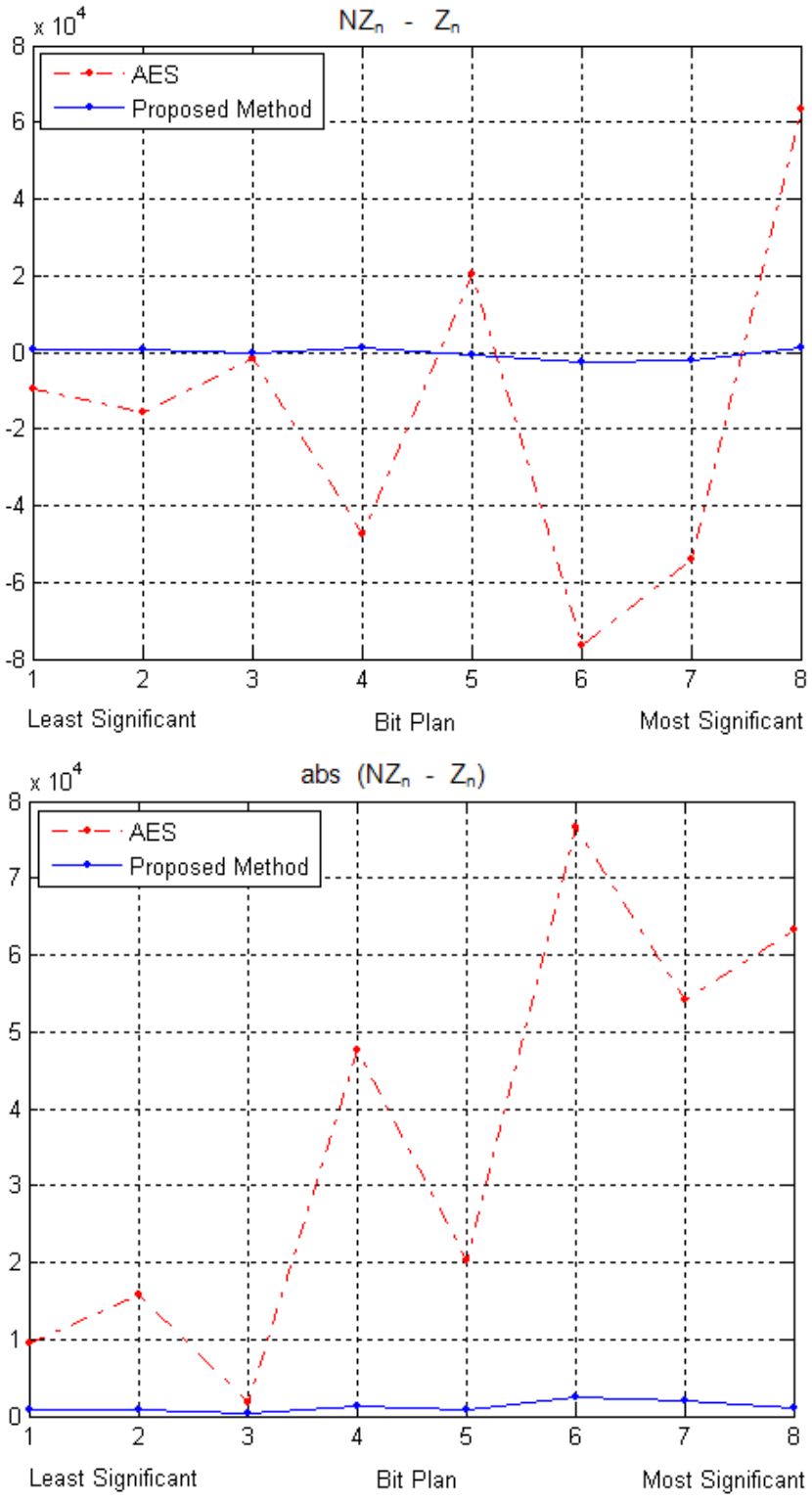


Fig.8. Monobit test on the encrypted images shown in Fig. 7. (top) the difference between the sum of Non-zero values (NZ_n) and Zero values (Z_n) across the 8 bit planes and (bottom) the absolute value (abs) of the above graph data.

Runs test: The focus of this test is the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits. The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow [40]. Testing this on image "*pepper_encrypted.bmp*" yields: *P-value=erfc((1048656-(2*2097152*0.4999*(1-0.4999)))/(2* sqrt(2*2097152)*0.4999*(1-0.4999)))=erfc(0.0782)= 0.9120.* The total number of runs for this example (i.e., *pepper_encrypted.bmp*) denoted by the value "*1048656"* is large enough to indicate an oscillation in the bit stream which is too fast as can be expected in a random sequence. Since the obtained *P-value* of 0.9120 is ≥ 0.01, we accept the sequence as random.

Cross-covariance sequence: This test estimates the cross-covariance sequence of random processes. A natural image tends to have different randomness along its bit eight levels as shown in Fig. 9. It is simply the cross-correlation of mean μ removed sequences as in Eq. 7.

$$\phi_{\varepsilon}(\mu) = E\{\varepsilon_i - \mu\} \quad (7)$$

where E{.} is the expected value operator.
The sequence is further normalized so the auto-covariances at zero lag are identically 1.0, i.e., the spike appearing at zero in Fig. 10.

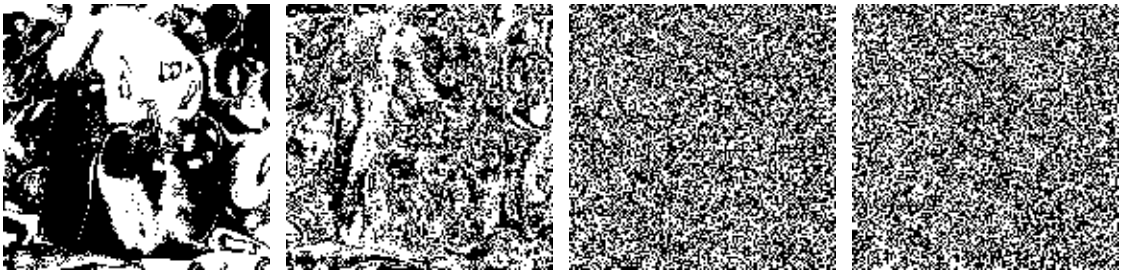
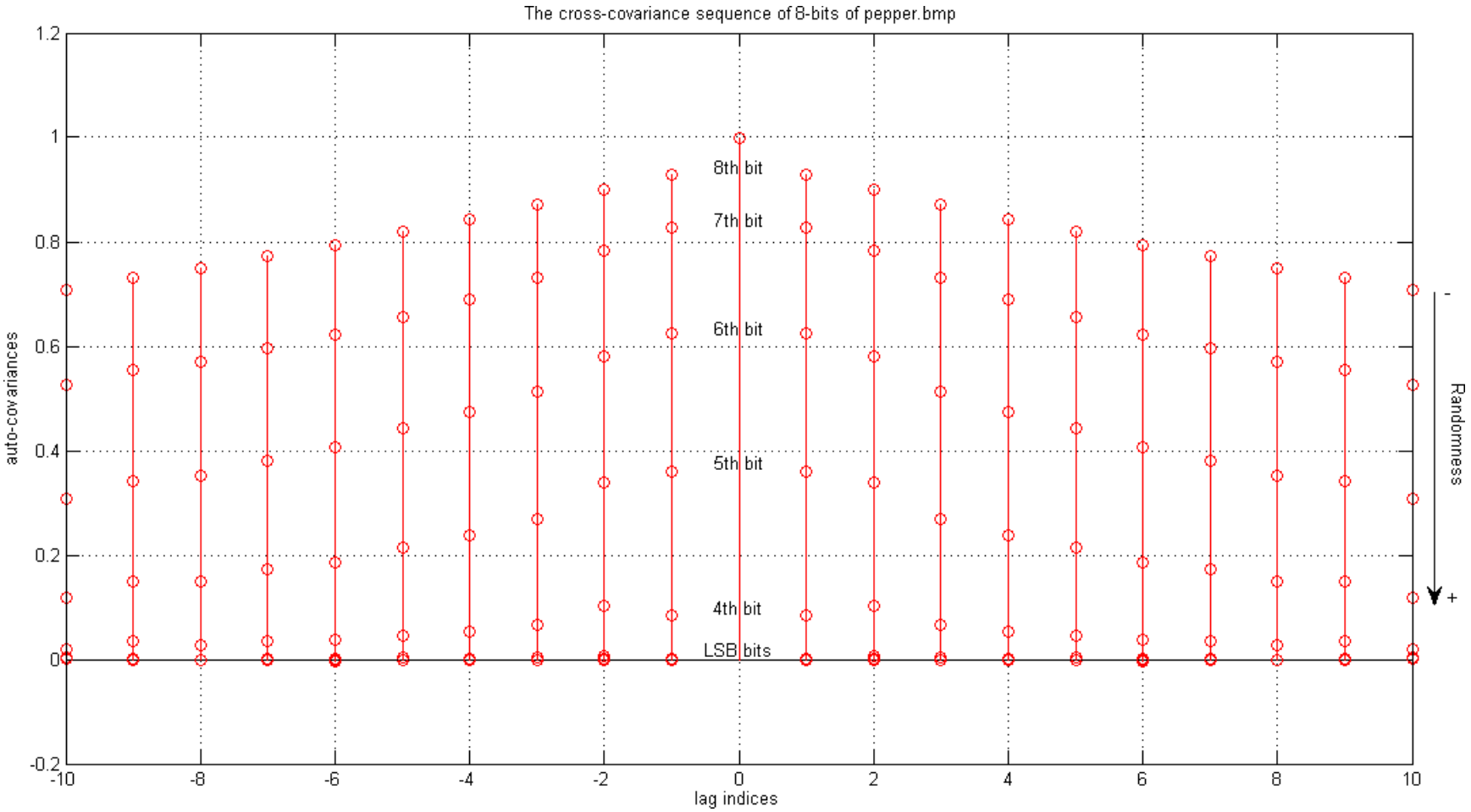


Fig. 9. Figure showing the randomness in natural images. (from left to right) original *pepper.bmp* 7th bit, 5th bit, 3rd bit and 2nd bit plan.


The cross-covariance sequence of 8-bits of pepper.bmp
auto-covariances
lag indices
8th bit
7th bit
6th bit
5th bit
4th bit
LSB bits
Randomness
-
+
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-10
-8
-6
-4
-2
0
2
4
6
8
10

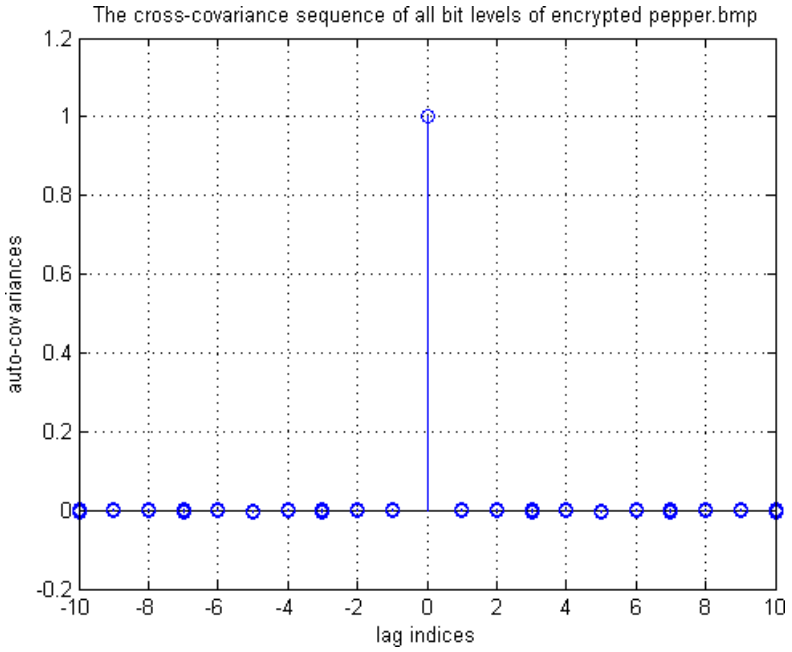


Fig. 10. Cross covariance test for randomness, (top) projection of each bit level from the plain image *pepper.bmp* and (bottom) a great randomness shown on all bit levels of the encrypted image. This phenomenon definitely helps mimic the least significant bits when embedding the encrypted secret data.

E. Differential Analysis

In order to determine the secret key, an adversary might try to establish a relationship between the plain image and its cipher version by observing the influence of a one pixel change on the overall encryption output. This kind of cryptanalysis becomes void when such a slight change results in a major transformation on the cipher. This influence is usually measured in percentage using the metric NPCR (Number of Pixel Change Rate) which calculates the number of pixel differences in two cipher images relating to two plain images having only one pixel difference and created using identical secret key.

Let the plain image's cipher be $\overline{\mathrm{A}}$ and the one pixel difference generated cipher be $\overline{\overline{\mathrm{A}}}$, then the NPCR can be obtained straightforwardly.

$$NPCR = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} D_{i,j}}{W \times H} \times \ 100\% \qquad (8)$$

Where, $D_{i,j} = \begin{cases} 0 & if \quad \overline{A}_{i,j} = \overline{\overline{A}}_{i,j} \\ 1 & if \quad \overline{A}_{i,j} \neq \overline{\overline{A}}_{i,j} \end{cases}$, $1 \geq i \leq H, 1 \geq j \leq M$, H and W denote the height and width of the image, respectively.

According to Kwok and Tang [44], the expected value of NPCR of two random images is estimated by:

$$\xi(NPCR) = (1 - 2^{-L}) * 100\% \qquad (9)$$

Where, L corresponds to the number of bits that represent a colour component, for grayscale images L=8 bits. Hence, it is sought that $\xi(\mathrm{NPCR}) = (1-2^{-8}) \times 100\% = 99.6094\%$. Table 3 contrasts the proposed method with other algorithms in terms of NPCR. Lena and Goldhill images of size 512*512 were used, where the plain images used to produce $\overline{\mathrm{A}}$ and $\overline{\overline{\mathrm{A}}}$ have only one bit difference.

Table. 3. Pixel difference between images encrypted using the same key.

NPCR (%)		
Algorithm	Lena	Goldhill
[47]	99.6700	99.4700
[48]	99.4700	99.2300
[49]	99.1300	99.0800
[50]	99.3300	99.2100
[44]	99.6024	Not reported
[6]	99.6094	Not reported
[51]	99.5200 (Average)	Not reported
[46]	99.60937 (Average)	Not reported
[52]	99.6096 (Average)	Not reported
Proposed	99.6113	99.5953

F. Other Security issues

Apart from the above performance of the proposed system, we highlight here two additional aspects of the method. The first feature is that the proposed scheme is capable of not just scrambling data but also it changes the intensity of the pixels which contributes to the safety of the encryption. For convenience Fig. 11 illustrates a cropped grayscale matrix of size 4x5 from a natural image along with its encrypted version. As can be appreciated from the figure, the algorithm combines the confusion and diffusion.

The second feature of the proposed algorithm is the unbiased handling of both gray scale and binary images. Methods involving chaos are special cases where they can be considered analogous to encryption, and that is when we have a binary plain image.

If an image contains homogenous areas large redundant data will surf and thwart the efficiency of encryption algorithms laying ground for a codebook attack. This is due to those consecutive identical pixels which lead to the same repeated patterns when a block cipher is used in the Electronic Code Book (ECB) mode [53]. Since the proposed algorithm is not block based, therefore, this kind of phenomenon does not occur.

176	179	176	176
177	175	175	175
177	175	177	177
177	177	180	180
176	176	177	177

236	159	167	106
246	56	163	150
60	212	77	216
38	71	218	191
189	37	60	19

Fig. 11. A 4x5 cropped plain patch from a natural image (left) and its encrypted version using the proposed algorithm (right). Notice how same gray values are encrypted differently, this irregularity is very important to hamper any attempt to reverse attack the algorithm.

An attacker cannot work backwards to deduce previous random values by observing the internal state of the algorithm. Attackers can also use computer clusters to break encrypted strings by predicting the output until enough entropy is obtained. This might work on text encryption using a dictionary attack, but as far as digital imaging is concerned, the computational prediction of such entropy that mimics the human visual system (HVS) is complex and vague; therefore its feasibility is questionable.

Chosen-plaintext attack (CPA) is an attack model in which an attacker is presumed to have the ability to encrypt a plain image to obtain its corresponding cipher. The purpose of this attack is to exploit weaknesses in the encryption algorithm in the hope to reveal the scheme's secret key as shown in Eq. 10.

$$A = A' \otimes (B' \otimes Map) \tag{10}$$

where A is the decrypted image, A' its cipher, B' is the attacker's encrypted neutral image (i.e., Fig.12 (c)), $\otimes$ is the XOR operation and Map is the key ($B' \otimes Map$ is shown in Fig. 12 (d)).

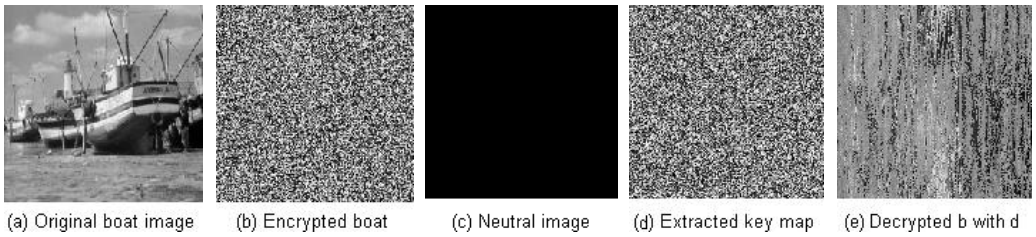


Fig. 12. CPA cryptanalysis attack. Note that (a) and (c) were encrypted with the same encryption key, to simulate the worst case in the attack. Then Eq. 10 is applied to yield (e).

V. RESULTS AND DISCUSSIONS

The results demonstrate that the algorithm is superior to the work of Pisarchik et al. [17] in terms of algorithm complexity and parameter requirements. Moreover, the algorithm is securely backed up by a strong 1D hash function. In [17] the desired outcome converges after some iteration which needs to be visually controlled to flag the termination of the program. However, in this work the algorithm is run only once for each colour component (R, G and B). The proposal needs only one input from the user (which is the password) and it will handle the rest of the process, while in [17] three parameters are required. The proposal obviously can be applied to gray scale images as well as binary images. These extensions are not feasible in [17] as they incorporate into their process relationships between the three primary colours (R, G and B). Finally, time complexity which is a problem admittedly stated in [17] would be reduced greatly by adopting this work's method. We coded the algorithm using MATLAB (which is an interpreted language) and Pisarchik et al. [17] used C#.

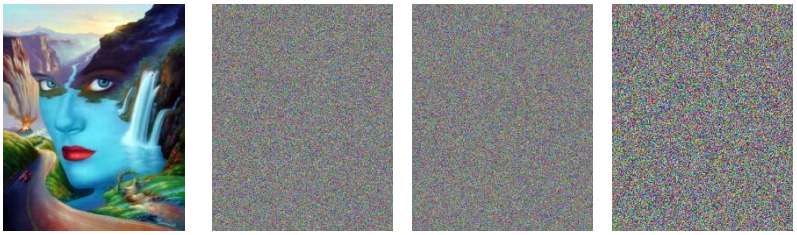
We tested the system on the same test image as in [17] to establish a fair judgement. To demonstrate visually the diffusion requirement being met, Fig. 13 illustrates the output with '*Steganography*' and with '*Steganographie*' as passwords. Even though only small change has occurred, the final two chaotic maps differ dramatically as can be seen from Fig. 13 (d). The proposed algorithm shows better performance compared to other recent methods such as the works in [5] [21] [41] [42] [43] [54], in addition to the conventional PRNG (see, Table 1).

Pisarchik et al. [17] measured the contrast between two given images by means of image histogram. Even though an image histogram is a useful tool; unfortunately, it does not tell us much about the structure of the image or about the displacement of colour values. Histograms accumulate similar colours in distinguished bins regardless of their spatial arrangements. A better alternative would be to use similarity measurement metrics such as the popular PSNR. PSNR values reach infinity if the two examined sets are identical. It is often expressed on a logarithmic scale in decibels (dB). Values falling below 30dB indicate a fairly low quality (i.e., distortion caused by embedding can be obvious); however, a high quality stego should strive for 40dB and above. Table 4 compares the PSNR values showing further information on the diffusion aspect. We have mentioned in section II that Pisarchik's algorithm [17]

involves a rounding operator applied each time the program is invoked by the different iterations.

Table 4. PSNR values of the different generated ciphers (unit measurement of PSNR is decibel (db)).

Chaos	Fig 13 (a)	Fig 13 (b)	Fig 13 (c)
Fig 13 (a)	-	7.8009 dB	7.8010 dB
Fig 13(b)	-	-	7.7765 dB



(a) (b) (c) (d)

Fig. 13. The 2D-SHA-2 algorithm: (a) test image (Mother of the Nature), (b) cipher using 'Steganography' as a password, (c) cipher using 'Steganographie' as a password and (d) the difference between *(b)* and *(c)*.

We do not adopt this feature as we believe there will be a loss of information when the embedded data is reconstructed.

Table 5 shows the advantages of using the proposed encryption method in comparison with other methods particularly in steganography applications. The algorithm is capable of surviving JPEG compression attacks up to 75%, below which the hidden data will be totally destroyed. We believe that surmounting JPEG compression was enhanced by the encryption of the payload since encryption often significantly changes the statistical characteristics of the original multimedia source, resulting in much reduced compressibility [8]. This resilience to attacks is deemed to be essential in image steganography or watermarking. In this case, the algorithm performs better than Peng's algorithm [55].

Fig. 13 shows on the left column Lena attacked with two kinds of noise, middle column decrypted Lena (AES) after noise attacks and right column decrypted Lena (proposed method) after similar attacks. It is very clear that the proposed algorithm is not merely an XOR operation since the PSNR is different than that obtained from original image (compare left column and right column).

Table 5. Comparison with different image encryption methods.

Method	Encryption matter	Steganography matters		
	Security	Balanced bit distribution	Tolerance to transmission faults	Suitability for image coding
AES/IDEA	excellent	weak, see Fig. 8	weak	average(*)
Chaos	average (see [23])	weak	average	very good
Bit stream ciphers	weak	very good	very good	very good
Proposed	good	very good	very good	very good

(*) If the message is encrypted with a block cipher and a given block size, the lengths of embedded data varies in multiples of this unit (Westfeld, the CRYSTAL project). See also, [6] [7] [53] [56]and Fig.13.

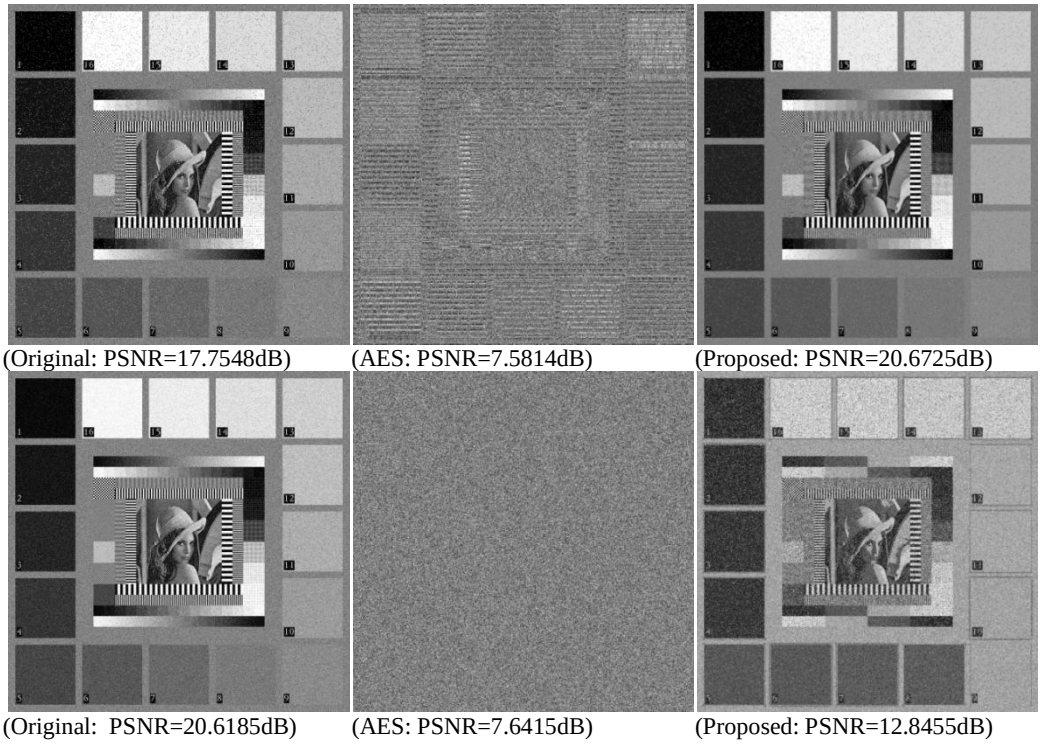


Fig. 13. Deciphered images after adding "Salt & Pepper" noise with 0.05 density (top) and "Gaussian" white noise of zero mean with 0.01 variance.

VI. CONCLUSION

We have presented in this paper a new encryption algorithm for two dimensional data such as images. The proposed algorithm is initiated by a password supplied by the user. Then we apply the extension of the SHA-2 algorithm to handle 2D data. An Irreversible Fast Fourier Transform (*IrFFT*) is applied to generate more scattered data. We have shown that the method outperforms several recent methods. A security analysis for the proposed scheme was presented. A comparison to other current systems was given, which shows the superiority of the proposal. We believe that very few of the current steganography/watermarking algorithms address in-depth the issue of data encryption prior to embedding. Those which do, rely heavily on the conventional encryption algorithms. This dependency is due, in part, to gaps in research literature pertaining to image encryption. The renowned block cipher algorithms, such as DES, AES, IDEA, etc, are not suitable for bulky data, i.e., digital images, due to their intensive computational process (see [4]) and probability of generating repetitive patterns. Recently the speed of such algorithms has however been boosted by hardware implementation.

This work bridges the extensive gap between cryptography and steganography and provides a neat answer to Westfeld's call for crypto-stego interaction (see section I). To this end we have developed a robust steganography method, with multiple layers of security to protect confidentiality, which acts as a useful application of the proposed cryptographic scheme, see [57,58].

The proposed method is not about XOR operation or a particular hash function, instead the novelty relies on the following points:

- It is mentioned in the article that a light weight stream cipher tailored to digital images (2D data with high correlation) and more precisely to digital image steganography/watermarking is needed. The latter require a continuous tone payload to be encrypted in such a way that produces a balances bit stream that would have a balanced flip effect on the cover image. This property is exhibited in Table 2 and the following figure.
- Transmission errors often occur either by natural faults or deliberately, e.g., added noise, image compression, therefore an encryption algorithm that is capable of surviving these attacks is needed too. This property is exhibited in Fig. 13.
- SHA2 has a very good property in that it is sensitive to initial values. In this paper we brought this sensitivity to work with the sensitivity of FFT of any changes in the spatial domain. FFT also produces a symmetric function out of any arbitrary signal, this symmetry is exploited to produce the balanced bit stream discusses earlier, see Fig. 6.
- Performance of the algorithm against various methods is shown in: Table 1 (against PRNG and other eight methods), Table 2 and Fig. 7&8&13 compare the proposed method against AES, Table 3 compares the NPCR of the proposed against nine algorithms, Table 5 summarises the good properties of the proposed method.

REFERENCES

[1] Online Software.
[S-Tools]: ftp://ftp.funet.fi/pub/crypt/mirrors/idea.sec.dsi.unimi.it/code/s-tools4.zip
[F5]: http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html

[2] H. Hioki, "A data embedding method using BPCS principle with new complexity measures," in *Proceedings of Pacific Rim Workshop on Digital Steganography*, 2002, July. pp. 30-47.

[3] CRYSTAL. (2004) Cryptography and Encoding in the Context of Steganographic Algorithms. [Online]. http://www1.inf.tu-dresden.de/~aw4/crystal/slides.slide_1.html

[4] K. Usman, H. Juzoji, I. Nakajima, S. Soegidjoko, M. Ramdhani, T. Hori, and S. Igi, "Medical image encryption based on pixel arrangement and random permutation for transmission security," in *Proceedings of IEEE 9th International Conference on e-Health Networking, Application and Services*, Taipei, Taiwan, 2007, 19-22 June. pp.244-247.

[5] M. Zeghid, M Machhout, L Khriji, A Baganne, and R Tourki, "A modified AES based algorithm for image encryption," *International Journal of Computer Science and Engineering*, 1(1)(2006) 70-75.

[6] V. Patidar, N.K Pareek, and K.K Sud, "A new substitution-diffusion based image cipher using chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, 14(7)(2009) 3056-3075.

[7] L.S. Chen and G.X. Zheng, "Chaos-based encryption for digital images and videos," in *Multimedia Security Handbook*, CRC Press, 2005, pp. 133-167.

[8] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *IEEE Transactions on Image Processing*, 15(7)(2006)2061-2075.

[9] Y. Wang, X. Liao, D. Xiao, and K.W. Wong, "One-way hash function construction based on 2D coupled map lattices," *Information Sciences*, 178(5)(2008)1391-1406.

[10] J. Wen, M. Severa, and W. Zeng, "A format-compliant configurable encryption framework for access control of video," *IEEE Trans. Circuits Syst. Video Technol*, 12(6)(2002)545-557.

[11] C.E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, 28(4)(1949)656-715.

[12] F. Shih, *Digital watermarking and steganography, fundamentals and techniques*. USA: CRC Press, 2008.

[13] D.C. Lou and C.H. Sung, "A steganographic scheme for secure communications based on the chaos and Euler theorem," *IEEE Transactions on Multimedia*, 6(3)(2004)501-509.

[14] Y. Wang, G. Ren, J. Jiang, J. Zhang, and J. Sun, "image encryption method based on chaotic map," in *Proceedings of IEEE 2nd Conference on Industrial Electronics and Applications (ICIEA)*, harbin, China, 2007, 23-25 May. pp.2558-2560.

[15] M. Ashtiyani, P.M. Birgani, and H.M. Hosseini, "Chaos-based medical image encryption using symmetric cryptography," in *Proceedings of IEEE 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2008)*, 2008, 7-11 April. pp.1-5.

[16] L. Bing and X. Jia-wei, "Period of Arnold transformation and its application in image scrambling," *Journal of Central South University of Technology*, Springer, 12(1)(2005) 278-282.

[17] A.N. Pisarchik, N.J. Flores-Carmona, and M. Carpio-Valadez, "Encryption and decryption of images with chaotic map lattices," *Chaos*, 16(033118)2006.

[18] A. Kanso and N. Smaoui, "Logistic chaotic maps for binary numbers generations," *Chaos, Solitons & Fractals*, 40(5)(2009) 2557-2568.

[19] P. L'Ecuyer, "Uniform random number generation," in *Handbook in Operations Research and Management Science*, Elsevier Science, 2006, vol. 13, p. 57.

[20] E. Solak and C. Çokal, "Comment on 'Encryption and decryption of images with chaotic map lattices," *Chaos*, 18(3)(2008)038101-038101-3.

[21] J. Zou, C. Xiong, D. Qi, and R.K. Ward, "The application of chaotic maps in image encryption," in *Proceedings of IEEE 3rd Northeast Workshop on Circuits and Systems NEWCAS*, Québec, Canada, 2005, 19-22 June. pp.331-334.

[22] F. Huang and Y. Feng, "Security analysis of image encryption based on two dimensional chaotic maps and improved algorithm," *Journal of Frontiers of Electrical and Electronic Engineering in China*, 4(1)(2009)5-9.

[23] C. Cokal and E. Solak, "Cryptanalysis of a chaos-based image encryption algorithm," *Physics Letters A*, 373(15)(2009)1357-1360.

[24] P. Refregier and B. Javidi, "Optical image encryption based on input plane and Fourier plane random encoding," *Optics Letters*, 20(7)(1995)767-769.

[25] U. Gopinathan, D.S. Monaghan, T.J. Naughton, J.T. Sheridan, and B. Javidi, "Strengths and weaknesses of optical encryption algorithms," in *Proc. 18th Annual Meeting of the IEEE Lasers and Electro-Optics Society*, 2005, 22-28 Oct. pp. 951-952.

[26] M. Singh, A. Kumar, and K. Singh, "Encryption and decryption using a sandwich phase diffuser made by using two speckle patterns and placed in the Fourier plane: Simulation results," *Optik - International Journal for Light and Electron Optics, article in press*, 2008, doi:10.1016/j.ijleo.2008.03.025 .

[27] M. Joshi, C. Shakher, and K. Singh, "Image encryption and decryption using fractional Fourier transform and radial Hilbert transform," *Optics and Lasers in Engineering*, 46(7)(2008)522-526.

[28] C.M. Shin and S.J. Kim, "Phase-only encryption and single path decryption system using phase-encoded exclusive-OR rules in Fourier domain," *Optical Review*, 13(2)(2006)49-52.

[29] A. Sinha and K. Singh, "A technique for image encryption using digital signature," *Optics Communications*, 218(4-6)(2003)229-234.

[30] L.H. Encinas and A.P. Dominguez, "Comment on 'A technique for image encryption using digital signature'," *Optics Communications*, 268(2)(2006)261-265.

[31] A. Sinha and K. Singh, "Reply to comment on 'A technique for image encryption using digital signature'," *Optics Communications*, 268(2)(2006)266-268.

[32] T. Gao and Z. Chen, "A new image encryption algorithm based on hyper-chaos," *Physics Letters A*, 372(4)(2008)394-400.

[33] R. Rhouma and S. Belghith, "Cryptanalysis of a new image encryption algorithm based on hyper-chaos," *Physics Letters A*, 372(38)(2008) 5973-5978.

[34] T. St. Denis, "Cryptography for Developers," in *Syngress*, 2006, pp. 203-250.

[35] N. Sklavos and O. Koufopavlou, "On the hardware implementations of the SHA-2 (256,384,512) hash functions," in *International Symposium on Circuits and Systems, ISCAS '03.*, vol. 5, 2003, pp. V-153 - V-156.

[36] I. Ahmad and A.S. Das, "Hardware implementation analysis of SHA -256 and SHA -512 algorithms on FPGAs," *Computers & Electrical Engineering*, 31(6)(2005)345-360.

[37] R. Glabb, L. Imbert, and G. Jullien, "Multi-mode operator for SHA-2 hash functions," *Journal of Systems Architecture*, 53(2-3)(2007)127-138.

[38] J.J.K. O'Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," in *Proceedings of IEEE International Conference on Image Processing (ICIP 97)*, vol. 1, Santa Barbara, CA, USA, 1997, pp. 536-539, 26-29 October. pp.536-539.

[39] A. J. Menezes, P.C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996, pp.175-177, ch. 5.

[40] A. Rukhin, 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 1, August 2008.

[41] K.W. Wong, B.S.H. Kwok, and C.H. Yuen, "An efficient diffusion approach for chaos-based image encryption," *Chaos, Solitons and Fractals*, 45(5)(2008)2652-2663.

[42] K.W. Wong, B.SH. Kwok, and W.S. Law, "A fast image encryption scheme based on chaotic standard map," *Physics Letters A*, 372(15)(2008)2645-2652.

[43] S. Lian, J. Sun, and S.Z. Wang, "A block cipher based on a suitable use of the chaotic standard map," *Chaos, Solitons and Fractals*, 26(1)(2005)117-129.

[44] H.S. Kwok and W. K.S. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, Solitons and Fractals*, 32(4)(2007)1518-1529.

[45] X. Tong, M. Cui, and Z. Wang, "A new feedback image encryption scheme based on perturbation with dynamical compound chaotic sequence cipher generator," *Optics Communications*, 282(14)(2009)2722-2728.

[46] S. Mazloom and A.M. Eftekhari-Moghadam, "Color image encryption based on Coupled Nonlinear Chaotic Map," *Chaos, Solitons and Fractals*, 42(3)(2009)1745-1754.

[47] A. Mitra, Y.V. Subba Rao, and S.R.M. Prasanna, "A new image encryption approach using combinational permutation techniques," *International Journal of Computer Science*, 1(2)(2006) 127-131.

[48] J.C. Yen and J.I. Guo, "A new chaotic key-based design for image encryption and decryption," in *Proc. of IEEE International Symposium on Circuits and Systems*, Geneva, Switzerland, 2000, May 28-31, pp. 49-52.

[49] S.S. Maniccam and N.G. Bourbakis, "Image and video encryption using SCAN patterns," *Pattern Recognition*, 37(4)(2004)725-737.

[50] D. Socek, S. Li, S.S. Magliveras, and B Furht, "Enhanced 1-D chaotic key-based algorithm for image encryption," in *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, 2005, 05-09 Sept. pp.406-407.

[51] C.K. Huang and H.H. Nien, "Multi chaotic systems based pixel shuffle for image encryption," *Optics Communications*, 282(11)(2009) 2123-2127.

[52] J. He, J. Zheng, Z.B. Li, and H.F. Qian, "An improved colour image encryption based on chaotic map and OCML model," in *Proceedings of International Conference on Networks Security, Wireless Communications and Trusted Computing*, China, 2009, 25-26 Apr. pp. 365-369.

[53] L. Shujun, C. Guanrong, and Z. Xuan, "Chaos-based encryption for digital images and videos," in *Multimedia Security Handbook*, Boca Raton,USA: CRC Press, 2004, pp. 133-167.

[54] X. Wang and J. Zhang, "An image scrambling encryption using chaos-controlled Poker shuffle operation," in *Proceedings of International Symposium on Biometrics and Security Technologies*, 2008, 23-24 April, pp. 1-6.

[55] Z. Peng and W. Liu, "Color image authentication based on spatiotemporal chaos and SVD," *Chaos Solitons and Fractals*, 36(4)(2008)946-952.

[56] J. Hu and F. Han, "A pixel-based scrambling scheme for digital medical images protection," *Journal of Network and Computer Applications*, 32(4)(2009)788-794.

[57] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A secure and improved self-embedding algorithm to combat digital document forgery," *Signal Processing*, 89(12)(2009)2324-32.

[58] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "A skin tone detection algorithm for an adaptive approach to steganography," *Signal Processing*, 89(12)(2009)2465-78.