Securing Information Content using New Encryption Method and Steganography

Abbas Cheddad, Joan Condell, Kevin Curran and Paul Mc Kevitt
School of Computing and Intelligent Systems, Faculty of Computing and Engineering
Londonderry, BT48 7JL, Northern Ireland, United Kingdom
Emails: {cheddad-a, j.condell, kj.curran, p.mckevitt}@ulster.ac.uk

Abstract

This paper proposes a novel encryption method with password protection based on an extended version of SHA-1 (Secure Hash Algorithm) that is able to encrypt 2D bulk data such as images. There has been a modest research in the literature on encryption of digital images though. The algorithm benefits also from the conjugate symmetry exhibited in what is termed, herein, an Irreversible Fast Fourier Transform (IrFFT). The proposed encryption method is a preprocessing phase which aims at increasing the robustness of Image Steganography against hackers. This scenario lays down a multi layer of security which forms a strong shield, against eavesdroppers, that is impossible to break. Both Shannon law requirements are met and results show promising results.

1. Introduction

Steganography, which is the science of concealing the very existence of data in another transmission medium, comes along not to replace Cryptography but rather to boost the security using its obscurity features. Steganography has various useful applications such as for Human rights organizations (as 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 and bank transactions to name few. Various algorithms were proposed to implement Steganography in digital images. We can categorize them into three major clusters, algorithms in the spatial domain such as *S-Tools* [1], algorithms in the transform domain for instance *F5* [2] and algorithms taking an adaptive approach combined with one of the former two methods for example *ABCDE* (A Block-based Complexity Data Embedding) [3]. Most of the existing Steganographic methods rely on two factors: the secrecy of the key and the robustness of the Steganographic algorithm.

This work aims at adding another unit of security which encrypts the secret image before the embedding process. Various hash algorithms are available such as MD5 (Message Digest 5), Blowfish, and SHA-1 (Secure Hash Algorithm 1) 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 [4]. Here, we attempt to extend SHA-1 (the terminology and functions used as building blocks to form SHA-1 are described in the US Secure Hash Algorithm 1 [5]) to encrypt 2D data. The introduction of Fast Fourier Transform (FFT) forms together with the output of SHA-1 a strong image encryption property.

This paper is organized as follows; section 2 discusses related work, followed by our proposal in section 3 and application to Steganography in section 3.1. Section 4 will analyzes and compares our results. Finally we conclude this work in section 5.

2. Related Work

Transferring images into chaotic maps was the obvious channel to the design of secure encrypted images. Chaos theory, which essentially emerged from mathematics and physics, deals with the behaviour of certain nonlinear dynamic systems that exhibit a phenomenon under certain condition known as chaos which adopts the Shannon requirement on diffusion and confusion [6]. Due to chaos' attractive features such as sensitivity to initial condition and random-like outspreading behaviour, chaotic maps become popular for data protection [4]. In the realm of 2D data, Shin [6] gave the following system in order to spread the neighbouring pixels into largely dispersed locations:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ l & l+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N \quad (1)$$

Where, $\det\left(\begin{bmatrix} 1 & 1 \\ l & l+1 \end{bmatrix}\right) = 1 \, or \, -1$, and l and N denote an arbitrary integer and the width of a square image respectively. W e referred to the determinant here as 'det'. After exactly 17 iterations the chaotic map converged into the original image as can be seen from Fig. 1.

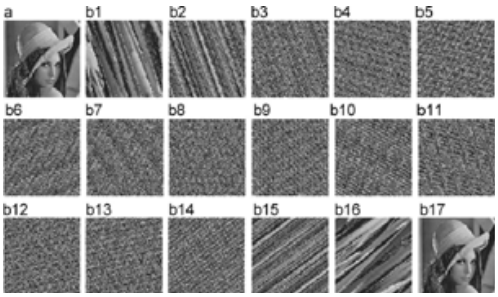

Figure 1. Different iterative results of applying Eq. 1 with *L*=2 on Lena image of size (101x101) [6].

Some remarks are worth noting here regarding this method:

A) The algorithm uses a determinant in its process, thus, the input matrix can only be square. A work around this problem might be in applying the algorithm on square blocks of a given image repetitively. However, it would generate noticeable periodic blocking artifact 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 reconstruction after some iteration, is also not an appealing factor. Given one of the iterations is used, if hackers have a prior knowledge of the algorithm and obtained the parameter "*l"*, which is actually not difficult to crack using brute force, they will be able to invest some time to add more iteration that will reveal the original image.

In a detailed and concise attempt to introduce image encryption, Pisarchik et al., [7] 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, 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 [7]. They introduced seven steps for encrypting images and seven steps for decryption. Their algorithm does not encompass any conventional hash algorithm, i.e., MD's family, SHA's family or Blowfish. Moreover, four parameters were used of which one was set constant and the other two were regulated. Their settings can have tremendous impact on the chaotic map quality as can be concluded from Fig 2 and Fig 3. Therefore, the receiver must know the decryption algorithm and the parameters beforehand. The algorithm is well formulated and adequately presented; it yields good results as proclaimed by the authors. The authors in [7] used a rounding operator which was applied recursively along the different iterations. An immediate concern will be about recovering the exact intensity values of the input image as the recovered image shown in their paper might be just an approximation because of the aforementioned operator. This is important, especially in the proposed application of Steganography where one wants to recover the exact embedded file rather than its approximation.

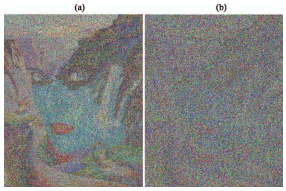

Figure 2. Colour sensitivity to number of cycles (a=3.9 and n=75). (a) Image encoded with j=1 and (b) j=2 [7].

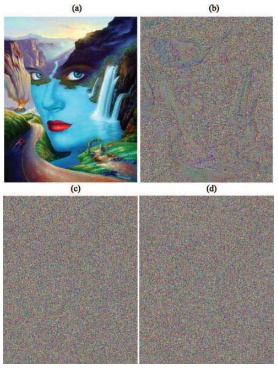

Figure 3. Colour sensitivity to number of iterations (a=3.9 and j=3). (a) Original image, (b) image encoded with n=1, (c) n=30 and (d) n=75 [7].

3. Proposed Method

The proposal exploits the strength of a 1D hash algorithm namely SHA-1 and extends it to handle 2D data such as images. The FFT is 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 exhaustive description of the algorithm step by step is illustrated in Fig 4 (Appendix). The proposal starts with a password phrase ***P*** supplied by the user to generate a SHA-1 based hash string ***H (P)***. The bit stream vector of ***H*** is then transformed to a matrix of fixed dimension 8x35. Parallel to this, the original image ***A*** is converted to a bit stream and reshaped to have the dimension of $8x(\prod(M,N))$. The key, herein 8x35, is short to accommodate the image bit stream. Therefore, the algorithm resizes the key towards the needed dimension, herein $8x(\prod(M,N))$. Obviously this step would result in repetitive patterns which would turn the ciphered image prone to attacks. To cope with this situation a modified *DCT* (Discrete Cosine Transform) followed by *FFT* are applied to provide the confusion and diffusion requirement and to tighten the security.

Let the resized key bit stream be $\lambda_{k,l}$ where the subscripts *k* and *l* denote the width and height after resizing the key respectively, i.e., 8, *M*N*. The FFT will operate on the DCT transform of $\lambda_{k,l}$ subject to Eq. 3.

$$f(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} F(x,y)e^{-2\pi i(xu+yu)/N} \qquad (2)$$
$$where, F(x,y) = DCT(\lambda_{k,l}), satisfying \quad Eq \quad 3$$

Note that for the transformation at the FFT and DCT levels the algorithm does not utilise the whole coefficients. Rather, it imposes the following rule, which generates at the end a binary random-like map. Given the output of Eq. 2 the binary map can be derived straightforwardly such that:

$$Map(x,y) = \begin{cases} 1 & iff \quad f(u,v) > 0 \\ 0 & Otherwise \end{cases} \qquad (3)$$

This map takes the positive coefficients of the imaginary part to form the ON pixels in the map. Since the coefficients are omitted the reconstruction of the password phrase is impossible, hence the name *Irreversible Fast Fourier Transform* (*IrFFT*). In other words, it is a one way hash function which accepts initially a user password. This map finally is *XORed* with the bit stream version of the image. The result is then converted into grayscale values then reshaped to form the ciphered image.

Another phenomena that was noticed worth exploitation is the sensitivity of the spread of the FFT coefficients to any kind of changes in the spatial domain, therefore if this is coupled with the sensitivity of SHA-1 algorithm to changes of the initial condition, i.e., Password phrase, the algorithm can meet easily the Shannon law requirements. For instance a small change in the password phrase will, with overwhelming probability, result in a completely different hash. The following exemplifies such assertion:

Input password: 'Steganography'
The corresponding Hash Function:
'40662a5f1e7349123c4012d827be8688d9fe013b'
Input password: 'Steganographie'
The corresponding Hash Function:
'c703bbc5b91736d8daa72fd5d620536d0dfbfe01'

So, the core idea here is to transform these changes into spatial domain where 2D-DCT and 2D-FFT are applied that introduce the aforementioned sensitivity to the 2D space. As such, images can be easily encoded securely with password protection.

3.1 Application to Steganography

After generating the chaotic map (encrypted image), the algorithm uses the colour transformation $RGB \rightarrow YCbCr$ on the cover image which will carry the encrypted data. The use of such a transformation is twofold; first to segment homogeneous objects in the cover image namely human skin region, and second to embed our data using the chrominance red (*Cr*) [8]. The *YCbCr* space can remove the strong correlation among R, G, and B matrices in a given image. This phenomenon is what interests the authors as less correlation between colours means less noticeable distortion if any alteration happens at this level. In this approach, the concentration on skin tone is motivated by some interesting applications of the final product. The algorithm starts first with the segmentation of probable human skin regions such that:

$$C = Bck \cup (\bigcup_{i=1}^{n} S_i), \qquad (4)$$
$$where: S_i \cap S_j = \varnothing(\forall i \neq j)$$

In Eq (4) *C* denotes the cover image, *Bck* background regions and (S_1, $S_{2,\ldots,}$ S_n) are connected subsets that correspond to skin regions.

Based on the carried experiments it was found that embedding into these regions produces less distortion to the carrier image compared to embedding in a sequential order or in any other areas. In addition to this, the algorithm yields a robust output against reasonable noise attacks and translation. The resistance to geometric distortion is feasible, unlike S-Tools and F5, since by selecting skin tone blobs the process can detect eye coordinates [9] which act as the reference points to recover the initial position and orientation and thus make the proposed method invariant to both rotation and translation. As shown in Fig 5 the proposed

algorithm was applied to digital image Steganography for two reasons, the first motivation is that embedding a random-like data into the Least Significant Bits (LSBs) would perform better than embedding the real natural data, secondly for security and fidelity reasons the embedded data must undergo a high encryption so even if it is accidentally discovered, which is unlikely to happen, the actual embedded data would not be revealed.

More specifically the study targets identification cards (IDs) which are prone to forgery in aspects pertaining to Biodata alteration or photo replacement. To protect photos, government bodies use a physical watermark on the photos using an iron stamp which is half visible or sometimes they use a normal stamp. This fragile shield of security can be easily deceived by mimicking the same stamp. The Biometric security measurement relies heavily on face features extraction and it is essential to have the system integrated into an external database with a real time connection to double check for identities. On the other hand, systems on chip are extremely expensive to roll out and require dedicated hardware. In addition, some chip circuits can be reverse engineered using a Radio Frequency Identification (RFID) technology. This occurred recently[1] in the Netherlands where two students from the University of Amsterdam broke the Dutch Public Transit Card. Such event brought the Netherlands' media into a storm of questionable security implementations.

This study proposes a cost effective yet highly secure system in which individuals' details are embedded into their photographs using a multi layer security channel reliant on Steganography. Steganography is the science of hiding the very existence of secret data in an innocuous way and the authors believe that this specific application of Steganography can overcome the difficulties mentioned earlier. Recently there have been a large scale losses of personal sensitive data in the UK e.g. loss of 25 million child benefit records after HMRC sent two unregistered/unencrypted discs to National Audit Office and the theft of laptop from Navy officer with personal details of 600,000 people. These incidents inspired another application of Steganography that is under investigation which aims at developing a highly secure large scale database. To evaluate the performance of the proposed system, a set of RGB images were used for this purpose. Fig 5 shows an example of the test data of which its PSNR (discussed in the next section) values are shown in Table 1. It is worth noting here that the frame appearing in Fig. 5 (top) is grabbed from a 16 sec duration video of size 2.76 MB and having 485 frames. The approximate embedding capacity would be about 34% of the total file size or 0.9407 MB.

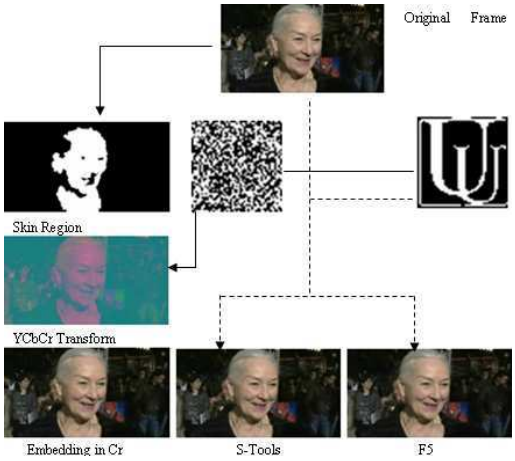

Figure 5. Figure showing the proposed Steganography system (Embedding in Cr) verses S-Tools and F5 algorithms.

Table. 1. Performance Comparison of the proposed method against S-Tools and F5.

Lady vs. UU		
	PSNR	Size Original /Stego-KB
Proposed	70.3956	315/222
S-T	69.5629	660/ 660
F5	48.9067	101/99.8

The authors believe that there will be numerous interesting applications for such an extended 2D-SHA-1 algorithm; however the concentration here is granted solely to Steganography.

4. Results and Discussions

This section reports the results which do better than the algorithm proposed by Pisarchik et al., [7] in terms of algorithm complexity and parameters requirement. Moreover, the algorithm unlike [7] is securely backed up by a 1D strong hash function. In [7], the desired outcome converges after some iterations which need to be visually controlled to flag the termination of the program; however, in the proposed case the algorithm is run only once for each colour component (R, G and B). The procedure needs only one input from the user which is the password and it will handle the rest of the process, while in [7] three parameters namely the reported a, j, and n need to be specified. The proposal obviously can be applied to gray scale images as well as binary images; these extensions are not feasible in [7] as

[1] Dutch Public Transit Card Broken, [online]. Available from: <http://www.cs.vu.nl/~ast/ov-chip-card/>, accessed on 02-03-2008 at 15:37.

they incorporate into their process relationship between the three primary colours (R, G and B). Finally, time complexity which is a problem admittedly stated in [7] would be reduced greatly by adopting our method; however, since MATLAB was used which is an interpreted language while [7] used C# for their application, this contrast was omitted here.

The algorithm was tested on the same test image described in [7] to establish a fair judgement. To demonstrate visually the diffusion requirement being met, Fig 6 illustrates the output with '*Steganographie*' 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 6 (d). This, combined with the sensitivity shown in Fig 6, will form excellent properties of the proposed algorithm. From Fig 6 it can be concluded that the proposed 2D encryption meets the diffusion requirement.

Pisarchik et al., [7] altered the test image by adding a black box at the lower right corner of the image and tried to visualise the difference by means of image histograms. Even though image histograms are a useful tool; unfortunately they, do not tell much about the structure of the image and in this case 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 Peak Signal to Noise Ratio (PSNR). PSNR values will run into infinity if the two examined sets are identical. PSNR is defined by the following system:

$$PSNR = 10\log_{10}\left(\frac{C_{\max}^2}{MSE}\right) \quad (5)$$

where *MSE* denotes the Mean Square Error which is given by:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy} - C_{xy}\right)^2 \quad (6)$$

and $C_{\max}$ holds the maximum value in the examined image, for example:

$$C_{\max} \leq \begin{cases} 1 \text{ in double precision intensity Images} \\ 255 \text{ in 8-bit unsigned integer intensity images} \end{cases}$$

x and y are the image coordinates, M and N are the dimensions of the image, S_{xy} is the original data and C_{xy} is the modified data.

Table 2 compares the PSNR values which inform us further with regards to the diffusion aspect.

Table 2. PSNR values of the different generated chaotic maps (unit measurement of PSNR is decibel (dB)).

Chaos	Fig 6 (b)	Fig 6 (c)
Fig 6 (b)	Inf	7.7765 dB
Fig 6 (c)	7.7765 dB	Inf

It was mentioned in section 2 that Pisarchik et al.'s algorithm involves a rounding operator applied each time the program is invoked by the different iterations. This study does not adopt this feature as it is believed that there will be a loss of information when the embedded data is reconstructed. In this proposal the algorithm goes in one direction and the recovery could be initiated by the same password and goes in parallel, i.e., not in the reverse order. Fig 8 depicts the input image and the recovered image, the PSNR reaches to infinity which means the two images are identical. Fig 9 shows the output of the algorithm applied to a binary image.

5. Conclusion

This paper presents a new encryption algorithm for two dimensional data such as images which is a pre-processing step in the ongoing research project named "Steganoflage" [10]. The proposal is initiated by a password supplied by the user. Then the process applies the introduced extension of the SHA-1 algorithm to handle 2D data. An Irreversible Fast Fourier Transform (IrFFT) is applied to generate a more scattered data. It was shown that the method outperforms that of Pisarchik et al., [7] in many ways.

References

[1] Brown, A. 1996. S-Tools [online]. [Accessed 04th April 2008]. Available from World Wide Web: <http://www.jjtc.com/Security/stegtools.htm>

[2] Westfeld, A. 2001. F5 [online]. [Accessed 04th April 2008]. Available from World Wide Web: <http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html>

[3] Hioki H. (2002). A Data Embedding Method Using BPCS Principle with New Complexity Measures. Proceedings of the Pacific Rim Workshop on Digital Steganography, pp.30-47.

[4] Yang, Wang., Liao Xiaofeng., Xiao Di., and Wong Kwok-Wo. (2008). One-way hash function construction based on 2D coupled map lattices. Journal of Information Sciences 178 (2008) 1391-1406.

[5] US Secure Hash Algorithm 1 [online]. [Accessed 05th April 2008]. Available from World Wide Web: <http://www.faqs.org/rfcs/rfc3174>.

[6] Shih, F. (2008). Digital Watermarking and Steganography, Fundamentals and Techniques. CRC Press. USA, pp: 22-24.

[7] Pisarchik A. N., Flores-Carmona N. J., and Carpio-Valadez M., (2006). Encryption and decryption of images with chaotic map lattices. Journal of CHAOS 16, 033118 (2006). The American Institute of Physics.
[8] Cheddad, A., Condell, J., Curran, K. and Mc Kevitt, P. (2008). Biometric Inspired Digital Image Steganography. In the proceedings of the 15th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'08). pp. 159-168. IEEE Computer Society.
[9] Cheddad, A., Mohamad, D. and Abd Manaf. A. (2008). Exploiting Voronoi Diagram Properties in Face Segmentation and Features Extraction. Pattern Recognition 41 (2008): 3842-3859, Elsevier Science.
[10] Steganoflage, available from: http://www.infm.ulst.ac.uk/~abbasc/

Appendix:

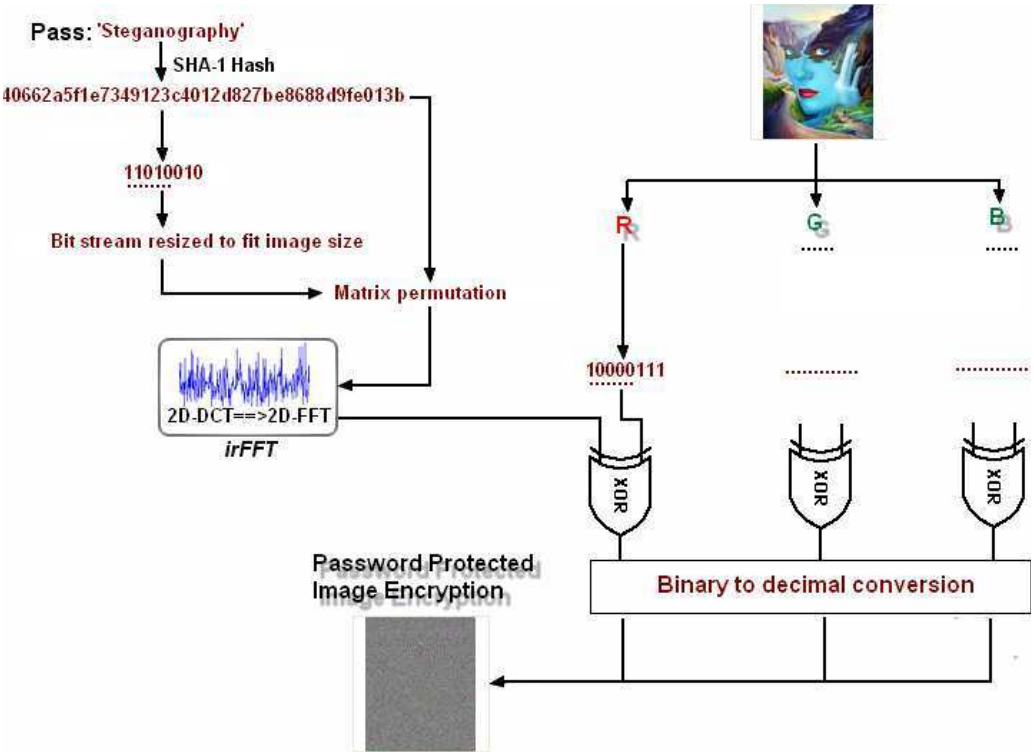

Figure 4. Block diagram of the steps used in the proposed algorithm for image encryption.

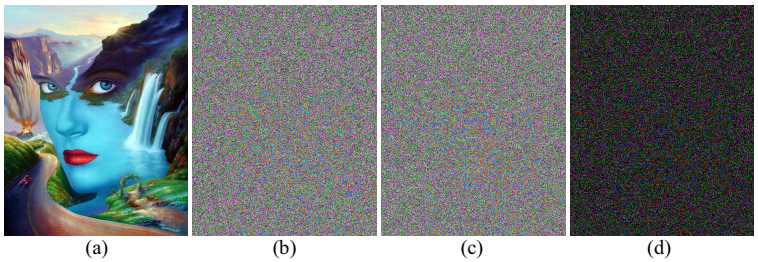

Figure 6. Our 2D-SHA-1 algorithm: (a) test image (Mother of the Nature), (b) chaotic map using '*Steganography*' as password, (c) chaotic map using '*Steganographie*' as password and (d) the difference between *(b)* and *(c)*.