

Hybrid Security System based on Wavelet Domain Watermarking and Chaotic Map Cryptography

Majdi Al-qdah

Faculty of Computers and Information Technology
University of Tabuk
Tabuk, Saudi Arabia

ABSTRACT

This paper presents a novel hybrid security system that combines wavelet based watermarking with block based chaotic maps cryptography. A watermark is chaotically encrypted before being embedded in the detail coefficients of the wavelet transformed cover image. Using the developed system, the potential of embedding more than one watermark inside one cover image is investigated. The results indicate that combining watermarking and cryptography gives an added layer of security. Also, even though embedding more than one watermark is more robust to attacks it can potentially distort small size cover images.

General Terms

Security, Data Hiding, Transforms.

Keywords

Watermarking, cryptography, chaos, Wavelets, multi-regions, attacks.

1. INTRODUCTION

Protection of information is essential to the daily transactions of all institutions: from e-mail, mobile phone applications and communications, and bank WebPages access. Protection can be in many forms; this research adopts more than one method of security: cryptography and watermarking.

Cryptography is a key technology for achieving information security in communications, computer systems, electronic commerce, and information systems[1]. It is an essential part of today's information systems. Cryptography can protect stored or transmitted information from unauthorized tampering and can prevent fraud in electronic commerce. It can prove and protect an individual's identity and protect Internet web pages and confidential documents of companies and government institutions.

Cryptography has two important concepts: confusion and diffusion. The purpose of confusion is to make the relation between a key and the cipher-image as complex as possible. Ciphers that do not offer much confusion are susceptible to frequency analysis. Diffusion spreads the influence of a single plain-image bit over many cipher-image bits. Changing a tiny part of the diffusion key should change each bit in the cipher-image with given probability [2].

There is a relationship between chaos and cryptography and many properties of chaotic systems such as: ergodicity, sensitivity to initial conditions/system parameters, mixing property, deterministic dynamics and structural complexity can be considered analogous to confusion, diffusion with small change in plain-image/secret key, diffusion with a small change within one block of the plain-image, deterministic pseudo randomness and algorithmic complexity properties of traditional cryptosystems [2]. There are number of ways through which digital chaos-based cryptosystems be realized: block ciphers based on forward and/or reverse iterations of

chaotic maps block ciphers based on chaotic round functions , stream ciphers implementing chaos-based pseudo random bit generators [2].

Encryption can be represented as $C = E(P, K_e)$, where K_e is the encryption key and $E(\cdot)$ is the encryption function; while decryption can be represented as $P = D(C, K_d)$ where K_d is the decryption key. The results of encryption is a cipher file while the results of decryption is a deciphered (original) file. In a block cipher, a group (blocks) of bits of fixed or variant length are encrypted block-by-block, and each block is mapped into another block with the same size, which differs from stream cipher, which encrypts the message pixel-by-pixel.

Several unique properties distinguish symmetric block encryption techniques based on two-dimensional chaotic maps from other symmetric block encryption algorithms. The main features are: a variable key length, a relatively large block size (several KB or more), and a high encryption rates. The cipher is based on two-dimensional chaotic maps, which are used for creating complex, key-dependent permutations. Unlike most today's symmetric encryption schemes, which rely on complex substitution rules while somewhat neglecting the role of permutations, the new cipher is based on complex permutations composed with a relatively simple diffusion mechanism. [3]

Chaos is used because it is deterministic (behaviors is known mathematically) and it is unpredictable (sensitive to initial conditions). Chaotic encryption scatters the pixels according to a given chaotic function. consider nonlinear and chaotic one dimensional maps $f: S \rightarrow S$ where S is a subset in R . The set $S = (0,1)$. The one- dimensional dynamical systems can be defined by a difference equation similar to

$$x_{k+1} = f(x_k), k = 0, 1, 2, 3, 4, \dots x_k \in S$$

where the variable k stands for time. A dynamical system consists of a set of possible states, together with a deterministic rule, which means that the present state can be determined uniquely from the past states. The orbit is the set of points, $\{x, f(x), f^2(x), \dots, f^n(x)\}$ where $f^2(x) = f(f(x))$ and $f^n(x)$ means n times iterating of the function $f(x)$. The starting point x for the orbit is called the initial value of the orbit. A chaotic orbit is one that forever continues to experience the unstable behavior that an orbit exhibits near a source, but that is not itself fixed or periodic [3].

On the other hand, watermarking has been an effective method for protecting the copyrights of rightful owners and authentication of documents; but watermarking must be robust and imperceptible in order to ensure copyright protection and Invulnerability to any possible attack. Any watermarking system consists of two main modules: embedding and extraction modules. In general, any embedding scheme tries to modify some features of a cover image, either changing pixel values directly or the altering

some coefficients of a transform before putting the image together into a watermarked image. Then the extraction process tries to correlate the watermarked image with the original image in order to extract/detect the watermark. In this paper, the cover image is a dyadic ($2^n \times 2^n$) size image.

watermarking works by inserting a watermark (image or text) into a document that can be authenticated by its rightful owner. Reversible data hiding is a technique to embed messages such as audio, video, or images into some distortionless cover media so that the original message can be extracted in a reversible way while keeping the cover content intact. Some algorithms purely do embedding and extraction in the spatial domain [4,5,6,7]. For example, in the difference expansion method in [5], differences between two adjacent pixels are doubled to generate a new least significant bit (LSB) plane for accommodating additional data. In [8] the image zero and peak values of the histogram are modified to embed the gray values of the image.

Some methods work when a transform is performed with a number of reversible data hiding transform methods have been proposed in recent years. For example, in [9], the cover data in the transform domain are encrypted and watermarked. The content owner encrypts a message using an encryption key while a data hider embeds messages using a data hiding key without knowing the original message content; a user will need an encryption key and data hiding key to decrypt and extract the original image message.

In [10], an invertible chaotic two-dimensional maps on a torus or on a square are adopted to create new symmetric block encryption schemes. The schemes are mostly suitable for encryption of large amounts of data, such as digital images or electronic databases. First, A chaotic map is generalized by introducing parameters and then discretized to a finite square lattice of points which represent pixels or some other data items. Although the discretized map is a permutation and thus cannot be chaotic, it shares certain sensitivity and mixing properties with its continuous counterpart as long as the number of iterations remains small. It is shown that for the two-dimensional baker map the permutations behave as typical random permutations. The discretized map is further extended to three dimensions and composed with a simple diffusion mechanism. As a result, a block product encryption scheme is obtained. To encrypt an $N \times N$ image, the ciphering map is iteratively applied to the image. In [11], Digital

watermarking algorithm based on chaotic encryption is proposed. The chaotic sequences are generated through chaotic cat map, which is determined by an initial condition and parameters. The watermark is added to the middle frequency coefficients of wavelet domain randomly by exploiting 2-D chaotic system, keeping the parameters of 2-D chaotic system as a private key. Experimental results demonstrated that the watermarking scheme is robust to typical attacks such as lossy JPEG compression, resizing and cropping.

In [12], an enhancement method of chaotic image encryption based on Baker's map is proposed. The enhanced symmetric-key algorithm supported a variable-size image as opposed to the algorithm which is mainly based on Baker's map that needed only square image for encryption. Also, the algorithm includes other functions such as password binding and pixel shifting for added security of the cipher image. The algorithm supports two modes of operation namely Electronic Code Book (EBC) and Cipher Feedback Chaining (CBC).

2. PROPOSED METHOD

This research combines image encryption with image watermarking. The proposed system uses logistic chaotic map to encrypt a given watermark image before inserting it into the detail (HL or LH) coefficients of the wavelet transformed cover image. The chaotic cipher depends on a private key obtained through a chaotic Pseudo Random Number Generators (PRNGs), which is represented by some initial floating point values. There is also a scrambling of pixels step before encryption using a scrambling key. If the watermark is large in size, then the encryption can be performed in a block by block fashion by splitting it into non-overlapping blocks, B_i such that:

$$\text{Size}(B_i) = Y_i, \sum_{i=1}^{BN} Y_i = M * N, \text{ where } BN \text{ is the number of Blocks.}$$

Then the encryption proceeds by applying masking between the image pixels' blocks and a generated chaotic block sequence for only one block. In this paper, a single block encryption is adopted. The overall system has two sides: a sender side (partitioning, scrambling, encrypting, then watermarking) and a receiver side (partitioning, extraction, decrypting, unscramble). Fig. 1, 2 show the sender and receiver sides of the system, respectively.

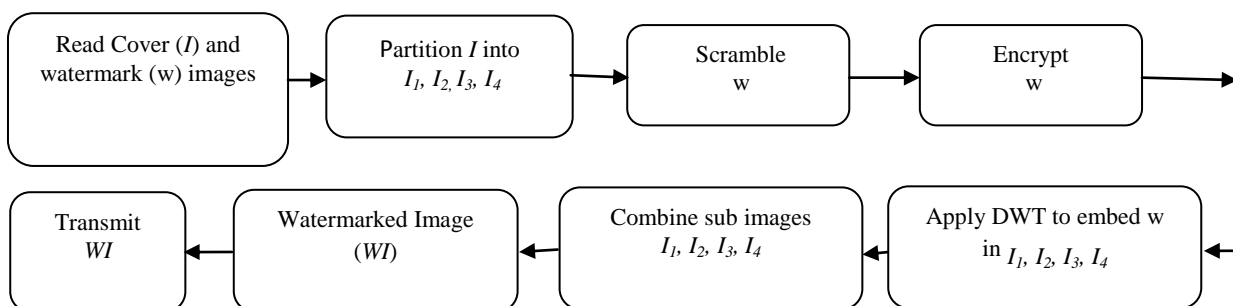


Fig. 1: Sender side

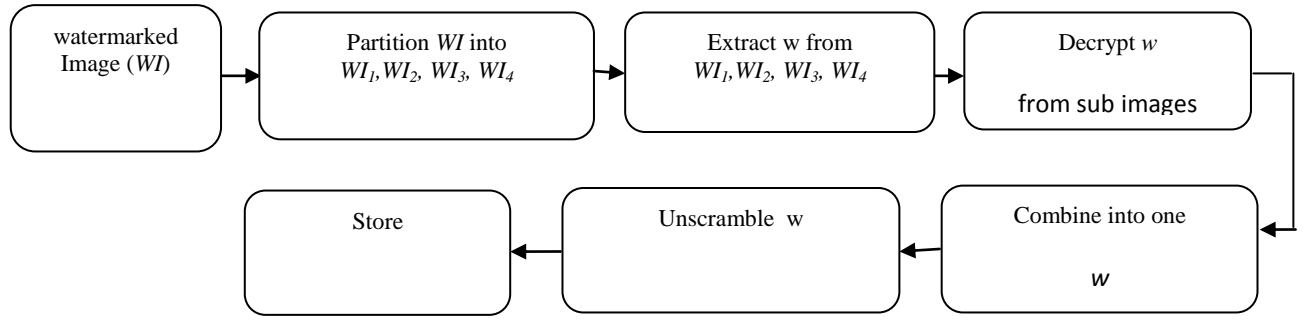


Fig. 2: Receiver Side

A. Steps of scrambling and encrypting:

Input: Watermark gray scale image (w)

Output: Cipher scrambled watermark image.

- Read the watermark image $w(i,j)$ of size $(m \times n)$
 $w = \{w(i,j): i=1,2,3,\dots,M, j=1,2,3,\dots,N\}$
- Scramble the image pixels with a permutated sequence using a minimum 15 randomly generated characters scrambling key.
- Convert the scrambled image array $(m \times n)$ into image vectors $(p1, p2, p3, \dots, pL)$; where $L=m \times n$.
- Choose a chaotic map defined by
 $x_{k+1} = \lambda x_k(1-x_k)$, $x_k \in (0,1)$, where λ is the control parameter (set to maximum chaos value) with x_0 as the initial condition.
- Generate a sequence of a chaotic points of the same size as the image vectors.
- Generate ciphering key sequences of chaotic decimal numbers, $(k1, k2, k3, \dots, kn)$
- Apply XOR to mask the image vectors and the generated key sequences.
- Combine the resultant ciphered vectors into a ciphered image of size $(m \times n)$

B. Steps of decrypting and unscrambling:

Input: Cipher and scrambled watermark image.

Output: Original watermark image (w)

- Read the cipher watermark image $w(i,j)$ of size $(m \times n)$
- Convert the image array $(m \times n)$ into image vectors $(c1, c2, c3, \dots, cl)$ where $l=m \times n$.
- Choose the same chaotic map used in encrypting the watermark.
- Generate a sequence of a chaotic points of the same size as the image vectors.
- Generate key sequences of chaotic decimal numbers, $(k1, k2, k3, \dots, kn)$
- Apply XOR between the image vectors and the generated key sequences.
- Combine the resultant original watermark vectors into the original watermark image of size $(m \times n)$

- Unscramble the image using the same scrambling key.

Next, a semi-blind watermarking is performed. After partitioning the cover image into four equal non overlapping segments (sub images), embedding is performed of the chaotically encrypted watermark in the horizontal (HL) or the vertical (LH) detail coefficient in each of the four non-overlapping discrete wavelet transformed (DWT) sub images. The entropy of each sub image's HL and LH component is calculated and the maximum of the two components (LH and HL) is chosen for embedding the watermark. The HH and LL components are not affected. The HH component is not changed since it contains edge related information of the image while the LL component contains the larger coefficients of the sub image. A random matrix is generated and embedded for each pixel value of the watermark; while the pixels are extracted using the correlation coefficient.

C. Steps for watermarking:

Input: Cipher watermark image and cover image.

Output: Watermarked image (Iw)

- Set the gain factor (α) to some integer value (2) for embedding.
- Read the cover image (I) of size $(2^n \times 2^n)$.
- Partition the cover image into 4 equal sub images (I_1, I_2, I_3, I_4) .
- Decompose each sub image into LL, HL, LH, and HH components using DWT.
- Calculate the entropy of the HL and LH sub bands of each sub image and set a flag for the maximum of the two. $E = -\sum(p * \log_{10}(p))$, p contains the count of the histograms.
- Read the ciphered scrambled watermark image (w) and convert it into a vector.
- Add a Gaussian noise PN- sequence to the maximum of HL and LH for each sub image when the pixel value of the watermark==0
$$I_{HL,LH} = 2 * \left(rand\left(\frac{2^n}{2}, \frac{2^n}{2}\right) - 0.5 \right) + HL, LH$$
- Perform IDWT to obtain a watermarked sub images (Iw_1, Iw_2, Iw_3, Iw_4) .
- Combine the four sub images into one watermarked image (Iw)
- Save the watermarked image.

D. Steps for extracting the watermark:

Input: Watermarked image (I_w)

Output: watermark ciphered image.

- Read in the watermarked image (I_w)
- Divide the watermarked image into four equal sub images ($I_{w_1}, I_{w_2}, I_{w_3}, I_{w_4}$).
- Decompose each sub image into LL, HL, LH, and HH sub bands.
- Read a flag to determine where the watermark had been embedded (HL or LH).
- Read in the original watermark (w) and the key.
- Convert watermark to a vector and initialize its values to one.
- Generate the same watermarking Gaussian noise PN-sequence.
- Find the correlation coefficient (ρ) between the random matrix and the embedding sub band.
- Store an extracted pixel when the correlation value > mean of correlation.
- Reshape the watermark vector back to an image.
- Perform a pixel majority (>2) of detection from each sub image to obtain a single watermark.
- Calculate the PSNR and MSSIM metrics for the detected watermark.

3. RESULTS

At the sender side, a binary watermark of size (64x64) image is scrambled and encrypted using the steps stated earlier. Fig. 3 shows the original, scrambled, and ciphered watermark images. Then, after partitioning a standard Barbara (512x512 at 8bits/pixel) cover image and applying the watermarking steps and inserting the ciphered scrambled watermark in each of the four parts of the divided cover image, four watermarked sub images are obtained. Next, the sub images are added together to form one watermarked image as fig. 4 shows. The process is reversed at the receiver side by doing the extraction and deciphering of the watermark. Fig. 4 shows four extracted deciphered unscrambled watermarks from each sub image and the combined weighted extracted watermark. A majority weight for each extracted pixel was taken (>2) meant a detection of a watermark pixel. The deciphering and unscrambling are an exact reversal of ciphering and scrambling shown in fig. 3.

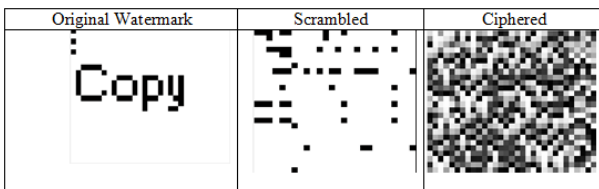


Fig. 3: Original, scrambled, and ciphered watermark

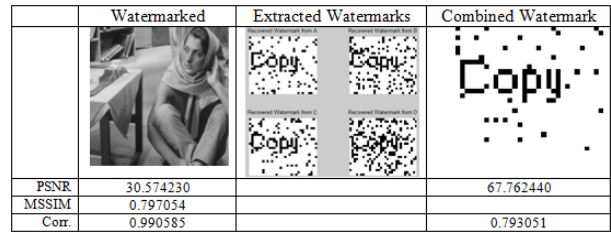


Fig.4: Watermarked Barbara image, four extracted watermarks, combined extracted watermark

For evaluation purposes, most image processing techniques use PSNR and MSE, shown in equation 1,2 respectively, to compare two images, where PSNR and MSE are calculated for an original image and the corresponding processed image.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \dots \dots \dots (Eq1) \quad [4]$$

$$MSE = \sum_{i=1}^N \sum_{j=1}^M \frac{(I_{ij}^d - I_{ij}^n)^2}{NM} \dots \dots \dots (Eq2) \quad [4]$$

where N,M denote the image size, $\{I_{ij}^d\}$ is the pixel value of the original image, and $\{I_{ij}^n\}$ denotes the pixel value of the watermarked image. On the other hand, the correlation coefficient measures the similarities and differences between the original image and the watermarked image or between the original watermark and the extracted watermark. The coefficient has a value between 0 and 1. A value of 1 means an exact or highly close match between the two images. The correlation coefficient is given by equation 3.

$$\rho = \frac{\sum_{i=1}^N w_i w_i'}{\sqrt{\sum_{i=1}^N w_i} \sqrt{\sum_{i=1}^N w_i'}} \dots \dots \dots (Eq3) \quad [13]$$

Where N is the number of pixels in the watermark, w_i is the original watermark, and w_i' is the extracted watermark.

Additionally, Human Visual System (HVS) metric called the Multi-Scale Structural Similarity Image Metric (MSSIM) is used. The MSSIM includes several peculiarities of the HVS with a range from 0 to 1. Larger values of MSSIM relate to a better visual quality. The structural similarity image quality paradigm is based on the assumption that the HVS is highly adapted for extracting structural information from the scene, and therefore a measure of structural similarity can provide a good approximation to perceived image quality [14].

The average PSNR value for the watermarked image is about 31 db, which is very comparable to the average value of 29.0 db when embedding a single watermark image in the cover image obtained in [13]; but the visual quality of the watermarked image is slightly distorted; The MSSIM visual quality index had a value of 0.797 indicating some visual distortion in the watermarked image. The extracted watermark had an average PSNR value of about 68 db and a correlation of above 80% between the extracted and the original watermark, which is a better value than the results obtained in [13] that used a single watermark for embedding.

In order to check the robustness of embedding few watermarks in a cover image, a series of attacks were inflicted on the watermarked image and the extraction procedure and the decryption procedure were followed. Fig. 5 shows the attacked images with various attacks: Gaussian noise, salt & pepper noise, cropping, wiener filter, intensity adjustment, Gaussian filter, and sharpening. The figure also shows the



PSNR and the correlation coefficient between the original and extracted watermark. Higher PSNR and correlation means better extraction and resolution of the extracted watermark and in affect better robustness to attacks.

	Attacked Image	Extracted Watermark
Gaussian Noise		
PSNR	20.58	60.34
Corr.	0.90	0.77



a) Gaussian Noise Attack ($m=0, \alpha^2=0.1$)

	Attacked Image	Extracted Watermark
Salt & Pepper		
PSNR	24.76	63.82
Corr.	0.96	0.88



b) Salt & pepper Attack ($D=0.01$)

	Attacked Image	Extracted Watermark
Cropping middle		
PSNR	9.9	64.15
Corr.	0.54	0.87



c) Cropping Attack (100:328, 100:328)

	Attacked Image	Extracted Watermark
Weiner Filter		
PSNR	9.34	57.16
Corr.	0.98	0.63


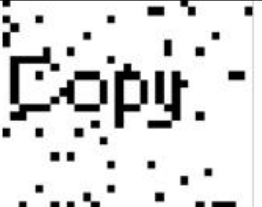
d) Weiner Filter

	Attacked Image	Extracted Watermark
Intensity Adjustment		
PSNR	9.36	62.11
Corr.	0.91	0.83

e) Intensity Adjustment ([0.2 0.4] to [0 1])

	Attacked Image	Extracted Watermark
Gaussian Filter		
PSNR	9.34	65.12
Corr.	0.99	0.91

f) Gaussian Filter Attack (size [3 3] and $\sigma = 0.5$)

	Attacked Image	Extracted Watermark
Sharpening		
PSNR	9.34	64.33
Corr.	0.85	0.89

g) 3x3 Sharpening Attack

Fig. 5: Attacked watermarked and extracted watermark images and their PSNR and Correlation coefficients under attacks (a) Gaussian Noise (b) Salt & Pepper Noise (c) Cropping (d) Weiner filter (e) Intensity Adjustment (f) Gaussian filter (g) Sharpening

As Fig. 6 and Fig. 7 show, multi-Region embedding has relatively high robustness since the recovered watermark after the attacks had an average PSNR value of around 63 db, which is a good value in comparison to embedding a single watermark. The extracted watermarks from all attacks are clearly visible except for the case when it was attacked with a Weiner filter. First, the watermarked images were attacked with Gaussian noise (mean = 0; variance =0.1) and the extracted watermark had a PSNR average value among all the images tested of 60 db and average correlation of about 0.77. Second, in attacking the watermarked images with salt & pepper noise (noise density (D) = 0.01), the extracted watermark had a PSNR average value of 64 db and an average correlation coefficient of 0.88. Third, in cropping the watermarked images with a middle square at pixel locations (100:328, 100:328), the extracted watermark had a PSNR average value of 64 db and average correlation coefficient of 0.87. Fourth, the wiener filter attack had the worst affect on the watermarked images; the extracted watermark had an average PSNR value of about 57 db and correlation of 0.63.

Fifth, the extracted watermark after adjusting the intensity from range [0.2 0.4] into [0 1] range; and cropping the values outside this range had a PSNR average value of about 62 db and average correlation coefficient value of about 0.83. Sixth, the Gaussian filter (size 3x3 and std = 0.5) attack resulted into a PSNR average value of about 65 db and correlation coefficient average value of about 0.91. Finally, the sharpening attack of size (3x3) resulted in an extracted watermark of average PSNR value of about 64 db and average correlation coefficient of about 0.89.

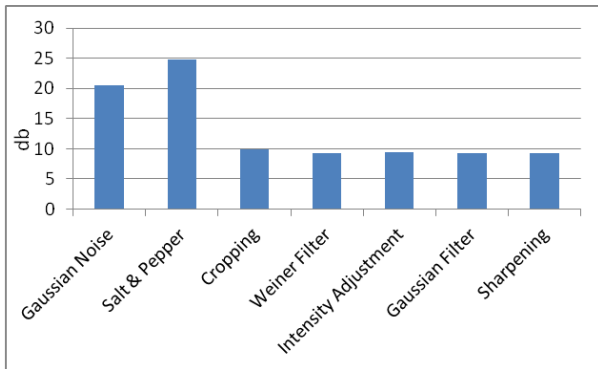


Fig. 6: PSNR of watermarked image after attacks

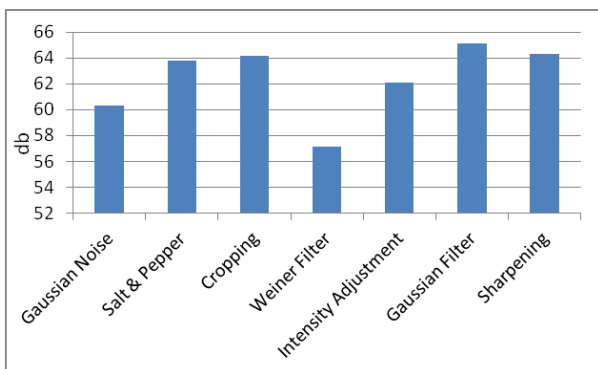


Fig. 7: PSNR of the extracted watermark after attacks

4. CONCLUSION

A combined security system of watermarking and encryption is proposed and implemented. Using the developed system, the embedding of few watermarks inside one cover image is investigated. Even though robustness is improved, it is fair to conclude that multi-region watermarking is more suitable for hiding a watermark inside an image for storage and transmission purposes rather than protecting copyrights of images since the visual quality of small size cover images suffers slightly when more than watermark is embedded.

5. REFERENCES

- [1] Dachsel, F. Schwartz W, "Chaos and Cryptography", IEEE Transactions on Circuits and Systems, 2001, vol. 48.
- [2] T. Habutsu, Y. Nishio, I. Sasase, and S. Mori, "A Secret key cryptosystem by iterating a chaotic map" in Proceeding Advances in Cryptology - EUROCRPT 91. Berlin, Germany: Springer-Verlag, 1991.
- [3] Jiri Fridrich, "Image Encryption Based On Chaotic Maps", Department of Systems Science and Industrial Engineering & Center for Intelligent Systems SUNY Binghamton, Binghamton, NY 13902-6000, USA, 1997
- [4] M. U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, "Lossless Generalized-LSB Data Embedding", IEEE Transactions on Image Processing, 14(2), pp. 253-266, 2005.
- [5] J. Tian. "Reversible Data Embedding Using a Difference Expansion." IEEE Transactions on Circuits and Systems for Video Technology, 13(8), pp. 890-896, 2003.
- [6] W. Hong, T. -S. Chen, Y. -P. Chang, and C. -W. Shiu, "A High Capacity Reversible Data Hiding Scheme Using Orthogonal Projection Error Modification", Signal Processing, 90, pp. 2911-2922, 2010.
- [7] C. -C. Chang, C. -C. Lin, Y. Lin, Y. -H. Chen, "Reversible Data Embedding Scheme Using Differences Between Original and Predicted Pixel Values", IET Information Security, 2(2). Pp 35-46, 2008.
- [8] Z. Ni, Y.Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding," IEEE Transactions on Circuits and Systems for Video Technology, 16(3), pp. 354-362, 2006.
- [9] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. G. B. Natale, and A. Neri, "A Commutative Digital Image Watermarking and Encryption Method in the Tree Structured Haar Transform Domain," Signal Processing: Image Communication, 26(1). pp. 1-12, 2011
- [10] Weiwei Xiao, Zhen Ji, Jihong Zhang, Weiyong Wu, "A watermarking algorithm based on chaotic encryption". Faculty of information engineering, Shenzhen University, Shenzhen, China, Proceeding of IEEE TENCON 2002
- [11] Huang-Pei Xiao Guo-Ji Zhang "An Image Encryption Scheme Based On Chaotic Systems", IEEE Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006.
- [12] Mazleena Salleh Suhariah Ibrahlim Ismail Fauzi Isnin, "Enhanced Chaotic Image Encryption Algorithm Based On Baker's Map", Department of Communication and Computer System, Faculty of computer Science and Information System, Universiti Teknologi Malaysia, Skudai 81 300, Johore, Malaysia. IEEE 2003
- [13] Harsh Varma, Abhishek Singh, Raman Kumar,"Robustness of Digital Image Watermarking Technique against brightness and rotation attacks", International Journal of Computer Science and Information Security (IJCSIS), vol. 1, no.1, 2009.
- [14] Zhou Wang1, Eero P. Simoncelli1 and Alan C. Bovik "Multi-Scale Structural Similarity for Image Quality Assessment", Proceeding of the 37th IEEE A silomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 9-12, 2003.