An Improved Self-Embedding Algorithm: Robust Protection against Lossy Compression Attacks in Digital Image Watermarking

Pratheepan Yogarajah, Joan Condell, Kevin Curran, Abbas Cheddad and Paul McKevitt

Abstract

Watermarking is the process of embedding watermarks into an image such that the embedded watermark can be extracted later. Lossy compression attacks in digital watermarking are one of the major issues in digital watermarking. Cheddad et al. proposed a robust secured self-embedding method which is resistant to a certain amount of JPEG compression. Our experimental results show that the self-embedding method is resistant to JPEG compression attacks and not resistant to other lossy compression attacks such as Block Truncation Coding (BTC) and Singular Value Decomposition (SVD). Therefore we improved Cheddad et al's. method to give better protection against BTC and SVD compression attacks.

Index Terms

Digital Watermarking; Compression Attacks; SVD; BTC; JPEG.

I. Introduction

Protecting digital image content is an important task in image security. To protect the content, digital image watermarking techniques are applied. In watermarking the secret information called the watermark, is invisibly embedded into the host digital image. A general watermarking framework for content protection is presented in [1].

Watermarking techniques can be categorized into two types (spatial and frequency domain) according to the embedding process. Watermarking in the frequency domain is more robust than watermarking in the spatial domain [2], because the watermark information can be spread out over the entire image [3]. Commonly used frequency domain transforms include the Discrete Wavelet Transform (DWT), the Discrete Cosine Transform (DCT) and the Discrete Fourier Transform (DFT). However, DWT [4] has been used in digital image watermarking more frequently due to its excellent spatial localisation and multi-resolution characteristics, which are similar to the theoretical models of Human Visual System (HSV) [5].

Y. Pratheepan, J. Condell, K. Curran, and P. McKevitt are with the School of Computing and Intelligent Systems, University of Ulster, UK.

A. Cheddad is with the Umea Centre for Molecular Medicine (UCMM), Umea Universitet, Sweden.

** This work was part funded by the Invest NI Proof of Concept (PoC) fund.

The content of watermarked digital images can be easily attacked by using image processing operations such as lossy compression. Invisible watermarking requires a reasonable robustness against compression attacks. Lossy compression algorithms tend to remove invisible information that can be related to the watermark. Watermark robustness under image compression is an essential issue for image content protection. Therefore, watermarks should combine invisibility and robustness simultaneously.

Recently Cheddad et al [6] proposed a method to protect the digital image itself using a secured robust self-embedding technique. In their method, a halftoned version (black and white image) of the original image is used as watermark. The calculated watermark is embedded in the 2D Haar DWT of the original image and the watermarked image is obtained. Then the Wavelet-based Inverse Halftoning via De-convolution (WInHD) is used on the extracted watermark from the watermarked image to recover the approximation of the original image. This is a blind watermarking scheme as the original image is not needed for the recovery process, see Figure 1.

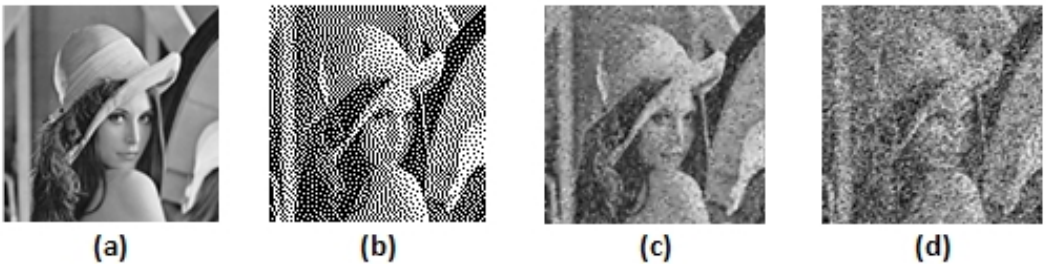

Fig. 1. (a) Original image, (b) halftoned binary image, (c) and (d) are recovered images from JPEG 95% and 85% quality compression attacks respectively.

JPEG 2000 is one of the modern lossy compression methods and is based on DWT. As the Cheddad et al. method is DWT based, it is resistant to JPEG compression attacks to a certain extent. They reported that their method is resilient to JPEG compression up to 80-75% [6]. There is no experimental results shown for other lossy compression techniques, such as Block Truncation Coding (BLC) and Singular Value Decomposition (SVD) etc. Our experimental results show that Cheddad et al's method is not robust to BTC and SVD lossy compression techniques. Therefore we improved Cheddad et al's method and experimental results prove that our method provides better recovery results on BTC and SVD compression attacks. A short version of this paper appears in the literature [7].

Our method is discussed in Section II. The secure image encryption algorithm is explained in section III. Sections IV and V explain the results and provide discussion and conclusions respectively.

II. Our Method

In DWT, an image is decomposed into a set of basis functions namely low frequency band (LL), high-low frequency band (HL), low-high frequency band (LH) and high frequency band (HH), see Figure 2(a). The low frequency band is a lowpass approximation of the original image and includes most energy of the image. The other bands include edge components of horizontal, vertical and diagonal directions at different scales and resolutions respectively.

According to the energy distribution, LL, is the most important. Hence in DWT domain, watermarks should be embedded in the low frequency band [8]. Cheddad et al. also selected the low frequency band of the 1^{st}-level 2D Haar DWT as their embedding area (i.e. LL_1) [6].

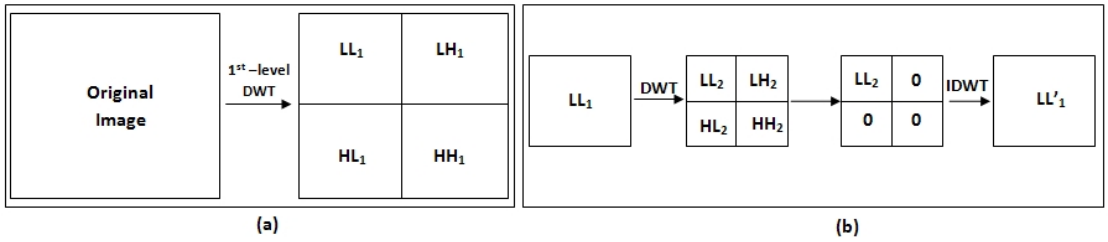

Fig. 2. (a) 1^{st}-level DWT decomposition and (b) LL_1' calculation.

Common image processing procedures, which watermarked images may encounter, such as data compression, low-pass filtering and subsampling, tend to change the low frequency coefficients less than high frequency coefficients [8]. Therefore it is necessary to reduce the high frequency coefficient effects to make the watermarked image resistant to compression attacks.

DWT has applications in image processing, where typically the approach is to DWT an image, alter the transform coefficients (by thresholding or zeroing), and inverse DWT to regain an altered image that has been de-noised, or its edges sharpened or blurred. The zeroing high frequency coefficient technique is applied to the digital image watermarking application in [9].

We also applied the zeroing technique to improve the performance of Cheddad et al.'s method against lossy compression attacks. In our method the original LL_1 is further wavelet transformed and then three high frequency bands (LH_2, HL_2 and HH_2, excluding LL_2) are initialised to zeros and inverse wavelet transformed. We then obtain another LL_1' from the process, see Figure 2(b). In our method, this newly calculated LL_1' is used for embedding instead of LL_1.

The watermark is embedded in LL_1' using an encryption algorithm explained in section III and the inverse DWT is applied to generate the watermarked image. The embedded watermark is extracted from the compression attacked watermarked image using a decryption algorithm (again see Section III). Finally the Wavelet-based Inverse Halftoning via De-convolution (WInHD) [11] is applied to the extracted watermark to recover the approximation of the original image.

Embedding the watermark into LL_1', instead of LL_1, may decrease the watermarked image quality, see Figure 3(b), but the extracted watermark reliability is increased, see Figure 3(d). The BTC and SVD compression attacks are applied to watermarked images. Brief information on BTC and SVD is given below.

Block Truncation Coding (BTC) is a lossy image compression technique. It divides the original images into blocks and then uses a quantizer to reduce the number of grey levels in each block while maintaining the same mean and standard deviation [12]. Sub blocks of 4x4 pixels allow compression of about 25%. Larger blocks allow

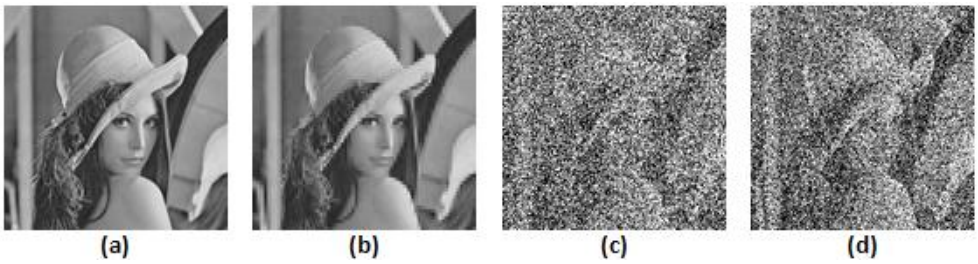

Fig. 3. (a) watermarked image using Cheddad et al's method, (b) watermarked image using our method, (c) recovered image from 4x4 block BTC compression attack using Cheddad et al's method and (d) recovered image from 4x4 block BTC compression attack using our method.

greater compression however quality also reduces with the increase in block size due to the nature of the algorithm.

Singular Value Decomposition (SVD) is one of the most useful tools of linear algebra. It is a factorization and approximation technique which effectively reduces any matrix into a smaller invertible and square matrix. Using (SVD) for image compression can be a very useful tool to save storage space [13], [14].

III. Image Encryption Algorithm

This algorithm is explained based on [6] and the encryption algorithm is fully described in [15]. A hash function is more formally defined as the mapping of bit strings of an arbitrary finite length to strings of fixed length [16]. 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, [17]) to encrypt digital 2D data. The introduction of Fast Fourier Transform (FFT) forms together with the output of SHA-1 a strong image encryption setting. Let the key bit stream be $\lambda_{k,l}$ where the subscripts k and l denote the width and height after resizing the key's bit stream respectively, i.e., 8, $M * N$, where M, N are the plain image's dimensions.

The FFT will operate on the DCT transform of $\lambda_{k,l}$ subject to Eq. 2.

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

where $F(x,y) = DCT(\lambda_{k,l})$ satisfying Eq. (2). Note that for the transformation at the FFT and DCT levels we do not utilise all of the coefficients. Rather, we impose the following rule, which generates at the end a binary random-like map. Given the output of Eq. 1 we can derive the binary map straightforwardly as:

$$Map(x,y) = \begin{cases} 1 & iff \quad f(u,v) > 0 \\ 0 & otherwise \end{cases} \tag{2}$$

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 function which accepts initially a user password. This map finally is XORed with the binary version of each colour component separately. The core idea here is to transform

these changes into the spatial domain where we can apply 2D-DCT and 2D-FFT that introduce the aforementioned sensitivity to the two dimensional space. As such, images can be easily encoded securely with password protection.

The watermark images are securely embedded using the encryption algorithm explained above. The decryption technique is also similar to the encryption algorithm and can be referred in [15].

IV. Results and Discussion

In this section, we illustrate and evaluate the performance of the proposed method against JPEG, BTC and SVD compression attacks on color and grayscale images[1]. The images, *Lena, Baboon, pepper* and *F16*, are used as the cover images for our experiments. Two different experiments are carried out to show the effectiveness of our proposed method against Cheddad et al.'s method.

In both experiments, the watermarked images are generated using Cheddad et al. and our method, see Figure 3(a) and (b). These watermarked images are attacked by JPEG, BTC and SVD compression techniques. The binary watermarks are extracted from attacked watermarked images and the approximation of the original image is recovered. For quantitative evaluation, the PSNR (Peak Signal-to-Noise Ratio) is introduced to evaluate the performance between the original image and recovered image. The PSNR is defined as follows:

$$\mathbf{PSNR} = 10log_{10}(\frac{255^2}{\mathbf{MSE}})dB \tag{3}$$

$$\mathbf{MSE} = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(a_{i,j} - b_{i,j})^2}{n * m} \tag{4}$$

where $m * n$ is the image size, $a_{i,j}$ and $b_{i,j}$ are the corresponding pixel values of cover and recovered images.

A. Experiment I

In this experiment, the 'UU-logo' grayscale image (64x64 pixels), Figure 4(a), is chosen as the secret image. This image is converted to a binary image, Figure 4(b), using dithering operation with WInHD [11]. Then this binary image is embedded in the cover image (i.e. *lena, baboon, pepper* and *F16*, 512x512 pixels) and embedded image, Figure 4(c), is generated. Then the embbeded image is attacked by lossy compression techniques, JPEG, SVD and BTC. A BTC 4x4 block compression attacked embedded image is shown in Figure 4(d).

Figure 4(e) shows the extracted watermarked image from the attacked embbeded image. Finally the grayscale image is reconstructed from the extracted watermark image using inverse halftoning via de-convolution (WInHD) [11], Figure 4(f). Then the reconstructed image, Figure 4(f), and the original secret image, Figure 4(a), are compared to see the robustness of Cheddad et al.'s method and our proposed method on image recovery against the lossy compression attacks.

Figure 5 shows the robustness of Cheddad et al.'s method and our proposed method against the SVD lossy compression attack with singular values 80, 100 and 160 respectively. Here the grayscale image, Figure 4(a), is used as the secret image (i.e. watermark) and the images *pepper, baboon, F16* and *Lena* are used as cover images, Figure 5(a).

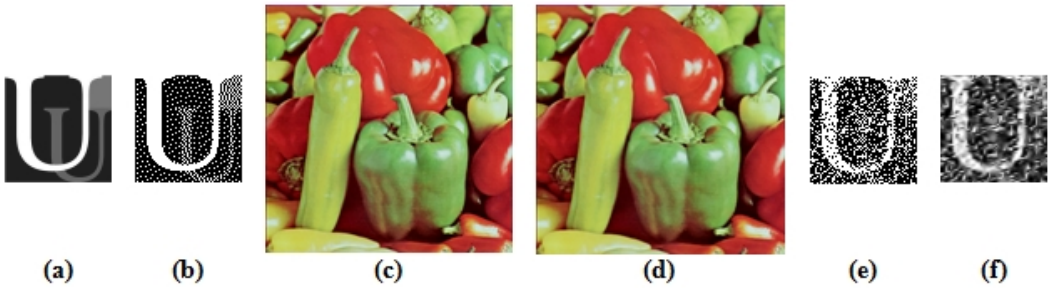

Fig. 4. (a) 'UU-logo' grayscale image, (b) watermark binary image from (a) using dithering operation with WInHD [11], (c) embbeded image, (d) attacked image using BTC 4x4 block compression attack, (e) extracted watermark and (f) reconstructed grayscale image.

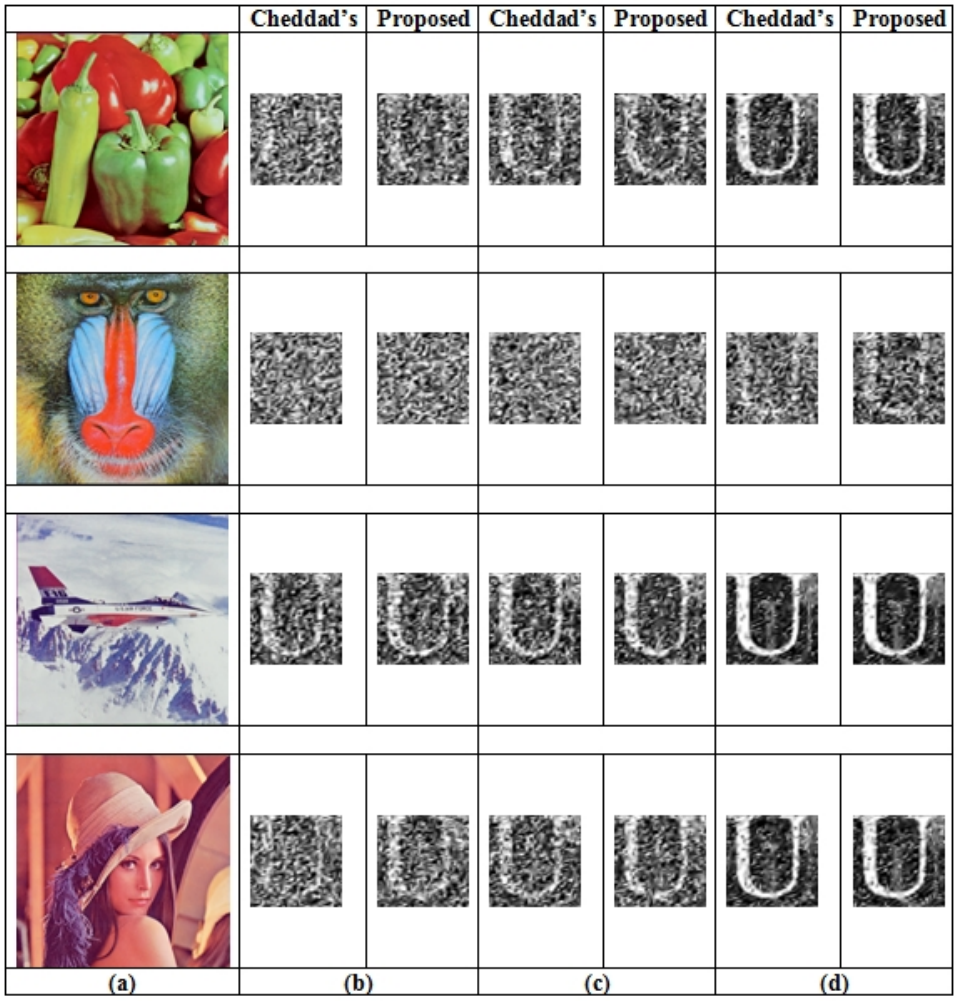

Fig. 5. (a) represents cover image. (b), (c) and (d) represent reconstructed secret images using dithering operation [11] from SVD compression attacks with singular values = 80, 100 and 160 respectively.

TABLE I

PSNR RESULTS - SVD LOSSY COMPRESSION ATTACK

Cover Image	Methods	Singular Values						
		60	80	100	120	140	160	180
Lena	Cheddad's	8.58	9.41	10.81	12.39	13.69	15.24	16.49
	Proposed	8.91	10.53	11.44	13.15	14.60	15.81	17.09
Baboon	Cheddad's	7.60	7.73	7.71	8.29	8.81	9.49	9.87
	Proposed	7.75	7.85	7.98	8.72	9.37	10.08	11.02
Pepper	Cheddad's	8.10	8.77	10.14	11.10	12.47	13.61	14.65
	Proposed	8.27	9.39	10.46	11.52	13.10	13.75	14.97
F16	Cheddad's	9.56	10.85	11.83	13.68	15.54	17.24	17.98
	Proposed	9.67	11.11	12.70	14.32	15.67	17.10	18.01

TABLE II

PSNR RESULTS - JPEG COMPRESSION ATTACK

Cover Image	Methods	Jpeg Compression Quality				
		75	80	85	90	95
Lena	Cheddad's	7.57	7.71	7.61	7.63	8.10
	Proposed	7.66	7.56	7.79	8.00	8.27
Baboon	Cheddad's	7.92	7.71	7.78	7.82	8.71
	Proposed	7.75	7.85	8.02	8.13	8.44
Pepper	Cheddad's	7.61	7.63	7.88	7.96	8.35
	Proposed	7.69	7.47	7.89	7.86	8.34
F16	Cheddad's	7.68	7.69	7.69	7.80	8.20
	Proposed	7.75	7.70	7.75	7.86	8.10

TABLE III

PSNR RESULTS - BTC COMPRESSION ATTACK

Cover Image	Methods	Block Size			
		2x2	4x4	8x8	16x16
Lena	Cheddad's	20.04	13.22	10.44	9.06
	Proposed	20.63	14.02	10.85	9.12
Baboon	Cheddad's	19.84	9.38	8.38	8.24
	Proposed	20.79	9.79	8.30	8.11
Pepper	Cheddad's	20.16	12.32	9.78	9.01
	Proposed	20.47	13.03	10.34	9.05
F16	Cheddad's	21.09	14.56	11.72	10.18
	Proposed	21.20	14.52	11.53	10.17

The full set of PSNR results for SVD, JPEG and BTC lossy compression attacks are given in Tables I, II and III respectively. The results show that our proposed method is more robust than Cheddad et al.'s method.

B. Experiment II

In this experiment, we protect the cover image itself from lossy compression attacks. In this self-embedding digital image content protection method, first the digital image, that needs to be protected against compressions attacks, is selected (say 256x256 'Lena' grayscale image). The 'Lena' image is shown in Figure 6(a). Then the Floyd's [10] error diffusion digital halftoning technique is applied on the 'Lena' image to obtain the watermark (i.e. a halftoned black and white image), see Figure 6(b). Then this black and white 'Lena' image is embedded in the original 'Lena' image. The results of robustness against the lossy compression attacks follow.

Fig. 6. (a), (b) and (c) represent cover image, binary image from (a) using dithering operation with WInHD [11] and embbeded image respectively. (d), (e) and (f) represent reconstructed grayscale images from SVD lossy compression attacks with singular values 140, 160 and 180 respectively.

1) JPEG Compression attack: From Figure 7 we see that both methods perform similarly against JPEG compression quality factors 75%, 85%, 90% and 95%.

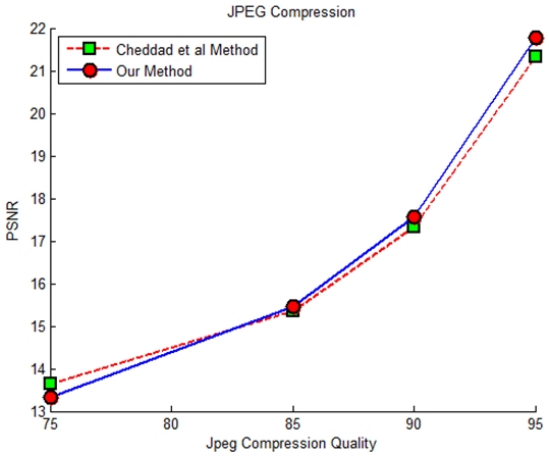

Fig. 7. PSNR results of JPEG compression attack on 'Lena' image.

[1] $http://www.petitcolas.net/fabien/watermarking/image\_database/index.html$

2) BTC Compression attack: Here 2x2, 4x4, 8x8 and 16x16 blocks are selected for experiments. When we apply 8x8 and 16x16 blocks BTC attacks on the watermarked image, the watermarked image acquires severely corrupted intensity of pixels. Therefore we could not see much difference in performance with 8x8 and 16x16 blocks BTC compression. Our method performed better when 2x2 and 4x4 blocks BTC compressions where applied to the watermarked images, see Figure 8.

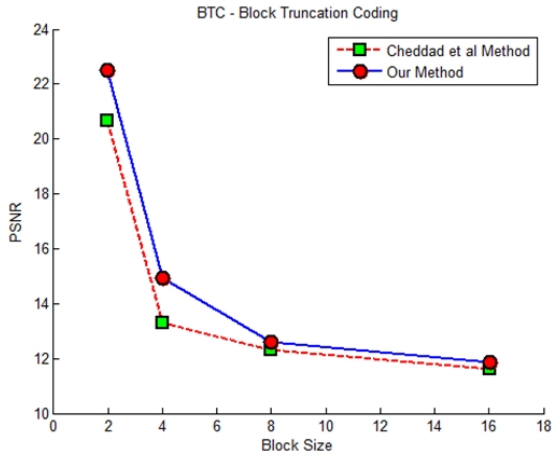

Fig. 8. PSNR results of BTC compression attack on 'Lena' image.

3) SVD Compression attack: Based on Figure 9, we can see that when 60 and 180 singular values are used for SVD compression attacks, the performances are similar for both methods. When 60 singular values are used, the recovered images from both methods are very noisy. At the same time, when 180 singular values are used, the quality of the recovered images from both methods are more similar. When the singular values between 60 and 180 are used, our method outperformes Cheddad et al.'s method.

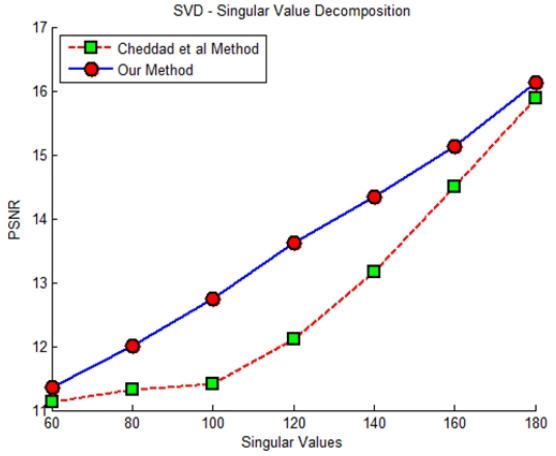

Fig. 9. PSNR results of SVD compression attack on 'Lena' image.

V. Conclusion

Lossy compression attacks in digital watermarking are one of the major issues when sending digital images over the internet. In this paper, we improved Cheddad et al's embedding method to make it resistant to lossy compression attacks such as JPEG, BTC and SVD. Our experimental results show evidence that the hidden content of the watermarked image can be recovered to a certain extent even though the watermarked image is attacked by lossy compression such as JPEG, BTC and SVD.

In our experiments, we considered hidden content as two different concepts. In experiment I, the hidden content is considered as a separate watermark image than the cover image. But in experiment II, the black and white conversion of the cover image is considered as the watermark image. We only experimented with grayscale images in experiment II. Future work will involve making our self-embedding method more applicable to a broader range of images, in particular colour images.

References

[1] Voyatzis G, Pitas I (1999) Protecting digital-image copyrights: a framework. IEEE. Computer Graphics and Applications. 19:18–24.

[2] Bender W, Gruhl D, Morimoto N, Lu A (1996) Techniques for data hiding. IBM Systems Journal. 35(3&4):313–336.

[3] Cox I J, Killian J, Leighton T, Shammon T (1997) Secure spread spectrum for multimedia. IEEE Transactions on Image Processing. 6(12):1673–1687.

[4] Vetterli M, Kovacevic J (1995) Wavelet and Subband Coding. Prentice-Hall, Engle-wood Cliffs.

[5] Wolfgang R B, Podichuk C I, Delp E J (1999) Perceptual watermarks for digital images and video. Proceedings of the IEEE, 87(7):1108–1126.

[6] Cheddad A, Condell J, Curran K, McKevitt P (2009) A secure and improved self-embedding algorithm to combat digital document forgery. Signal Processing 89(12):2324–2332.

[7] Pratheepan, Y., Condell, J.V., Curran, K., Cheddad, A., Mc Kevitt, P. (2010) An Improved Self-Embedding Algorithm: Digital Content Protection against Compression Attacks in Digital Watermarking. In Proceedings of The Second International Conference on Image Processing & Communications (IPC), 84:59–66.

[8] Daren H, Jiufen L, Jiwu H, Hongmei L (2001) A DWT-based image watermarking algorithm. IEEE Int Conf Multimedia and Expo. 429–432.

[9] Joo S, Suh Y, Shin J, Kitkuchi H (2002) A New Robust Watermark Embedding into Wavelet DC Components. ETRI Journal. 24(5):401–404.

[10] Floyd R W, Steinberg L (1976). An adaptive algorithm for spatial grayscale. Proc. Soc. Inf. Disp. 12:55–77.

[11] Neelamani R, Nowak R, Baraniuk R G (2002). WInHD: wavelet-based inverse halftoning via deconvolution. submitted to IEEE Trans. Image Process. for publication.

[12] Chanda B, Dutta Majumder D (2000). Digital Image Processing and Analysis. Prentice-Hall.

[13] Richards D, Abrahamsen A (2001). Image compression using singular value decomposition. linear algebra applications.

[14] Prasantha H S, Shashidhara H L, Balasubramanya Murthy K N (2007) Image Compression using SVD. Proc. of International Conference on Computational Intelligence and Multimedia Applications. 143–145.

[15] Cheddad A, Condell J, Curran K, McKevitt P (2010) A Hash-based Image Encryption Algorithm. Opt. Comm. Elsevier Science. 283(6):879–893.

[16] Wang Y, Liao X, Xiao D, Wong K (2008) One-way hash function construction based on 2D coupled map lattices. Inf. Sci. 178(5):1391–1406.

[17] US Secure Hash Algorithm 1, (2001) http://www.faqs.org/rfcs/rfc3174.