American Journal of Applied Sciences 11 (7): 1128-1134, 2014
ISSN: 1546-9239

doi:10.3844/ajassp.2014.1128.1134 Published Online 11 (7) 2014 (http://www.thescipub.com/ajas.toc)

PERFORMANCE ANALYSIS OF IMAGE SECURITY BASED ON ENCRYPTED HYBRID COMPRESSION

[1]Ramkumar, D. and [2]I. Jacob Raglend

[1]Department of ECE, Theni Kammavar Sangam College of Technology, Theni, India
[2]Department of EEE, Noorul Islam Centre for Higher Education, Nagercoil, India

Received 2014-02-02; Revised 2014-03-27; Accepted 2014-04-26

ABSTRACT

In this research, we propose an image security scheme using hybrid compression techniques. In this scheme, the data is being provided two-fold security by both encryption stage and hiding stage. The data/message which has to be secured undergoes encryption technique at the initial stage. In this stage, the permutation algorithm is employed which requires a pair of numbers as a key to permute the original message. Following the encryption stage, the deformed message is then embedded onto a JPEG image by considering the low and high quantization tables. The main motivation behind this research work is to provide image security through compression. The final result is an encrypted and compressed JPEG image with a different image quality. The receiver has to perform the reverse process to extract the original data/information. The performance analysis is performed in terms of PSNR for different quantization tables.

Keywords: Image Security, JPEG, PSNR, Quantization Table, Encryption, Permutation

1. INTRODUCTION

In recent years, signal processing in the encrypted domain has attracted considerable research interest. As an effective and popular means for privacy protection, encryption converts the ordinary signal into unintelligible data, so that the traditional signal processing usually takes place before encryption or after decryption. However, in some scenarios that a content owner does not trust the processing service provider, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. For instance, when the secret data to be transmitted are encrypted, a channel provider without any knowledge of the cryptographic key may tend to compress the encrypted data due to the limited channel resource.

Using lossy compression technique (Kumar and Makur, 2009), an encrypted gray image can be efficiently compressed by discarding the excessively rough and fine information of coefficients generated by orthogonal transform. A receiver may reconstruct the principal content of original image by retrieving the values of coefficients in the compressed image. Based on the homomorphic properties of the primary cryptosystem, the discrete Fourier transform in the encrypted domain can be implemented.

Similarly, in a buyer-seller watermarking protocol (Memon and Wong, 2001), the seller of digital multimedia product encrypts the original data using a public key and then permutes and embeds an encrypted fingerprint provided by the buyer in the encrypted domain. After decryption with a private key, the buyer can obtain a watermarked product. This protocol ensures that the seller cannot know the buyer’s watermarked version while the buyer cannot know the original version.

In this study, we are to propose an encryption and embedded algorithm for hiding the data over another image called the Host image. The algorithm makes use of Quantization process at the hybrid compression level and a single compressed image is obtained. The decoder decompresses the embedded image and extracts the original image (with the same resolution) by applying the data hiding key. After the encryption process, the pixels

Corresponding Author: Ramkumar, D., Department of ECE, Theni Kammavar Sangam College of Technology, Theni, India

are completely hidden so that intruders are unable to find any statistical information about the original image. Thus, this algorithm provides a high range of security.

2. RELATED WORKS

Zhou *et al*. (2006) presented an encryption methodology, named, Half tone visual encryption scheme. Their paper encodes a Secret binary Image (SI) into portions of random binary patterns. If the portions are zeroed onto transparencies, the secret image can be visually decoded by superimposing a qualified subset of transparencies. In Wang (2009) used region Incrementing Visual Cryptography Algorithm (IVCA) for a highly secured system. They have used MATLAB for the simulation of IVCA. Their results showed that the method consumed high elapsed time.

Iwamoto (2012) proposed a weak security notion for Visual Secret Sharing (VSS) design. VSS schemes were designed to be secure against attackers' eyesight, under such a weak security notion. Lian *et al*. (2007) achieved image security through the watermarking algorithm. They used visible watermarking in their scheme to acquire data security, which they named, 'Error Diffusion Watermarking Algorithm' and their experimental results proved a better performance.

Celik *et al*. (2005) pointed out the LSB data embedding methodology for data encryption and decryption. Their paper achieved 78% of PSNR only which leads to less recognition of images.

Several works have been proposed based on the compression of encrypted images. When a sender encrypts an original image for privacy protection, a channel provider without the knowledge of a cryptographic key and original content may tend to reduce the data amount due to the limited channel resource. A composite signal representation method packing together a number of signal samples and processing them as a unique sample is used to reduce the complexity of computation and the size of encrypted data. Johnson *et al*. (2004), the compression of encrypted data is investigated with the theory of source coding with side information at the decoder and it is pointed out that the performance of compressing encrypted data may be as good as that of compressing non-encrypted data in theory.

3. PROPOSED SECRET INFORMATION SCHEME

The standard JPEG scheme is mostly employed to deal with color images in RGB format. The first step is to convert the image from RGB space into luminance/chrominance spaces Y, Cb and Cr. Color space conversion step is followed by the sub-sampling step, where typically the chrominance channels (Cb and Cr) are sub-sampled with a rate equal to half the rate of the Y channel. All the channels are then partitioned into 8×8 non-overlapping blocks. The pixel values in each channel are shifted from the range [0,255] to [-128,127] for the next step Discrete Cosine Transformation (DCT). DCT is a powerful transformation which separates the low frequency and the high frequency coefficients, i.e., the low frequency coefficients are split and placed in upper left corner of the 8×8 block. Hence, the high frequency coefficients are reduced at this step, by applying specific Quantization Tables (QT). Larger the coefficient set in a quantization table, higher is the compression rate. But the image quality is significantly reduced and vice versa. Various softwares like MATLAB, Photoshop, etc. and various camera models use different QTs for the same image quality.

3.1. The Working of Proposed Message Hiding Scheme

Let us consider an example where a set of coefficients C_c being quantized by a factor Q_c. The set of DCT coefficients of size 8×8 contains C_{ij} from C_{11} to C_{88} mapping to a Quantization Table (QT) of the same size (i.e., 8×8). Every DCT coefficient is then quantized by the subsequent values present in the QT and it is rounded-off to the nearest integer, as given by Equation 1:

$$\hat{C}_{ij} = \text{round}\left(\frac{C_{ij}}{Q_{ij}}\right) \tag{1}$$

The DCT coefficient set is re-calculated as $\hat{C}_c = \hat{C}_{ij} \times Q_{ij}$ and then the value of $\hat{C}_c$ set is quantized for again for the second time by a factor C_q, which provides the DCT coefficients set Q_q after it is reconstructed. The difference between C_q and C_c will be a minimum value only when $Q_c = Q_q$ except for the case of $Q_q = 1$ in which no more quantization is possible. In simple, if C_c is already quantized by a factor Q_0, such that, $Q_0 > Q_c$ it should be understood that it is treated with a lower quality and the difference attains a minimal value as given by:

$$\text{difference} = \sum_{ij}\left[C_c^{ij} - C_q^{ij2}\right]^2 \tag{2}$$

where, i = j = 1,2,…8. Moreover, the value of difference in Equation 2 attains a local minimum at $Q_q = Q_0$. In order to embed the secret message onto a JPEG image, we need some specific regions of the JPEG image to be compressed with a lower quality Q_0. Those regions possessing lower quality are used for carrying the secret data/information and thus, this embedded image is used as a medium for secret communication. The operation of this algorithm is being illustrated in **Fig. 1**.

3.1.1. Encryption Stage

Figure 1 illustrates the operation of our proposed scheme. The secret image which is to be embedded is first encrypted by passing it through the encryption stage. Here an automorphism algorithm is applied for the permutation of the pixels in the secret message. Indeed, any encryption algorithm with the key being strong enough can be used at this stage. After some modulo operation is applied on the image, it turns out into a random pattern. Consider for example, after applying the modulo operator over the original image 66 times (n = 66), with parameter ω = 2 we get the deformed image similar to the original image; ω is the parameter to change the divisor in this modulo operation and is regarded as the decryption key. For recovery of the original image, (ω, n) must be known. For decoding, the same operator has to be applied with ω = 2 and (n = 126); therefore this operation has to be repeated 192 times to recover the original image. Therefore, for more security, different sets of (ω, n) can be used as the key required by the recipient to reconstruct the hidden image.

3.1.2. Data Embedding Stage

Embedding is the process of hiding the encrypted pattern onto a natural image like a mask. All the pixels of the secret image are embedded onto a specific region of a JPEG image using the quantization tables. Consider a secret message S of size X×Y S being binary image with S(x,y) = 0 for black pixels and S(x,y) = 1 for white pixels. Assume the host image H_o of size L×M. In order to maintain the shape of the secret image unaffected, the aspect ratios of the secret image and the host image should be identical, i.e., L/M = X/Y. The embedding process abides the following procedure for hybrid compression.

Initially, the secret image is split into equal blocks of size as per aspect ratio format and size of source image, which is to be encrypted and compressed. Then check each pixel values in secret image. If this pixel is black or ‘0’, then compress or quantize source image using the quantization table QT1. If the pixel is white or ‘1’, then compress or quantize source image using the quantization table QT2.This process is called as hybrid compression. Hence, this compression technique is based on two different quantization tables such as QT1 and QT2.

After hybrid compression is completed, the information/data which has to be hidden onto the compressed image is embedded in to hybrid compressed image using data hiding key. The data hiding key is nothing but the starting position of the hiding scheme of data in to the hybrid compressed image. If once this process is completed, then the final image is stored using high quality factor. The pseudo code for hybrid compression is explained in **Fig. 2**.

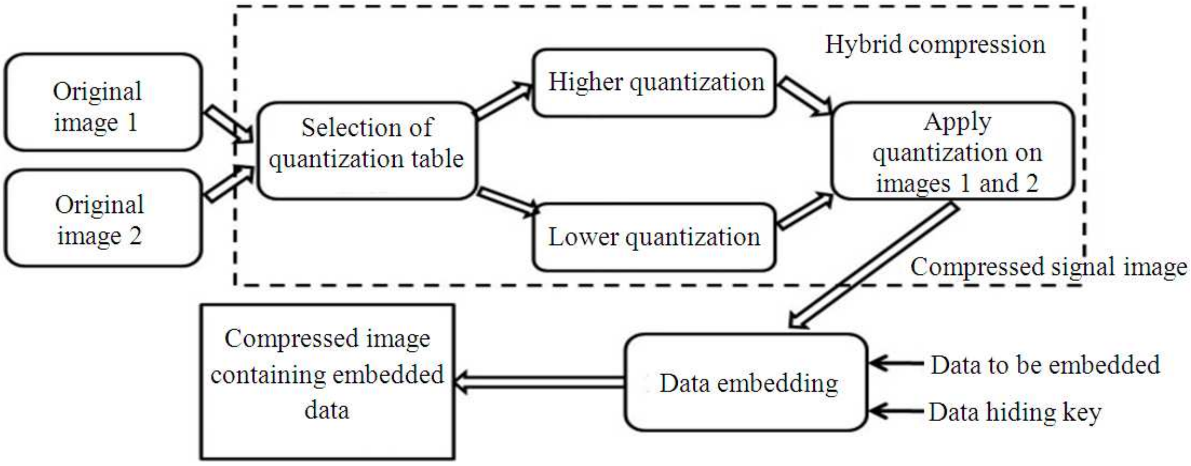

Fig. 1. Encryption and embedding at the transmitter side

3.1.3. Decoding/Decryption Stage

Here in decryption phase, the regions with different quality factors are first detected by the receiver to extract the pattern of the message embedded over the host image. As the message pattern is extracted, the recipient can rearrange the pattern with the help of decryption key already provided by the sender to restructure the hidden secret message from the host image. The block diagram of decoder is shown in **Fig. 3**. Let us consider a method in which black pixels are embedded in the lower quality regions. The recipient resaves the received JPEG image l_1 with lower quality (Q_2) to produce a new image l_2. The next step is to find the difference image (l_1-l_2). In the difference image, the compressed regions with lower quality are represented as black pixels, whereas other regions appear as white pixels. After subtraction, the embedded regions are considered to possess low values, i.e., almost 0. Hence, even if scaling is done, these regions must appear very dark. The following procedure is performed to recover the embedded data as well as original source image.

Initially, the hybrid stored compressed image is taken and restored as low quality factor. The embedded data is extracted from this low quality hybrid compressed image using encryption key as a key input to the recovery of data. After the data recovery process is over, we generate the difference image by subtracting higher quality hybrid compressed image from lower quality hybrid compresses image. Then, divide this image in to sub blocks of equal size.

```
Input:  Source and hidden image.
Output: Hybrid compressed image.

Upon receiving source and hidden image
If the file is available then
Check aspect ratio
If the aspect ratio is satisfied
Begin
If the pixel is '0' then quantize block using QT1
If the pixel is '1' then quantize block using QT2
End
If the hybrid compression is done then
Embed data using encryption key
End
If data embedding completed successfully then
Store the final image using high quality factor
End
```

Fig. 2. Pseudo code for hybrid compression

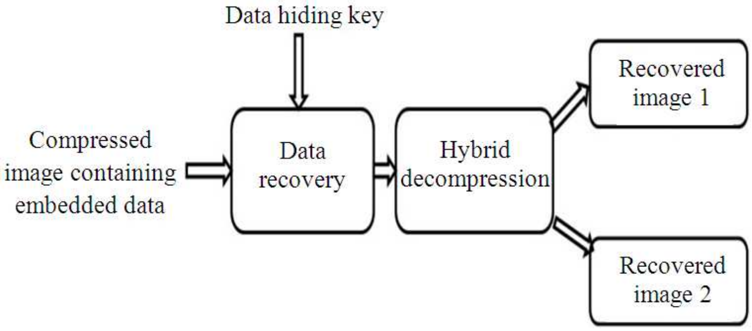

Fig. 3. Recovery of original images at decoder

```
Input:  hybrid compressed hidden image.
Output: Recovered data and source image.

1. Upon receiving hybrid compressed image
2. If the file is available then
3. Begin store this image using low quality factor.
4. End
5. If the encryption key is received then
6. Recover the data using key
7. End
8. If data recovery completed successfully then
9. Generate difference image
10. Find threshold
11. Decode as white or black pixels with respect to
    threshold
12. End
```

Fig. 4. Pseudo code for hybrid decompression

The smallest and largest values in each sub-block are evaluated and these values are noted as x1 and x2. The threshold value is determined by taking average of x1 and x2. If the sum of the pixel values in each block is less than the threshold, a black pixel is decoded; otherwise a white pixel is decoded. After this final step, the source image is recovered. The pseudo code for hybrid decompression is explained in **Fig. 4**.

4. RESULTS

For our experimental results, we have considered a host image of size 1024×1024 and the secret image to be hidden of size 128×128, after the permutation stage and embedded image, respectively. The performance of the proposed algorithm is analyzed in terms of PSNR, which is calculated by comparing the host image with embedded image. The performance parameter PSNR is given by the following Equation 3:

$$\text{PSNR} = 10 * \log 10(255^2 / \text{MSE})\text{dB} \tag{3}$$

The other parameter for performance analysis is Compression Ratio (CR) and it can be evaluated by the following expression Equation 4:

$$\text{CR} = \frac{\text{Original Image size}}{\text{Compressed Image size}} \tag{4}$$

The above equation states that the original image size is considered as source image before compression and compressed image size is considered as source image after compression.

Table 1. Quality table for quantization of black pixels

QT1 =	[1	1	1	1	1	1	1	1;
	1	1	1	1	1	1	1	1;
	1	1	1	1	1	1	1	2;
	1	1	1	1	1	1	2	2;
	1	1	1	1	1	2	2	3;
	1	1	1	1	2	2	3	3;
	1	1	1	2	2	3	3	3;
	1	1	2	2	3	3	3	3]

Table 2. Quality table for quantization of white pixels

QT1 =	[6	4	4	6	10	16	20	24;
	5	5	6	8	10	23	24	22;
	6	5	6	10	16	23	28	22;
	6	7	9	12	20	35	32	25;
	7	9	15	22	27	44	41	34;
	10	14	22	26	32	42	45	37;
	20	26	31	35	41	48	48	40;
	29	37	38	39	45	40	41	40]

In this experiment section, we have used two quantization tables, namely, QT1 and QT2. Among these two quantization tables, the Quantization **Table 1** (QT1) has lower coefficient values and Quantization **Table 2** (QT2) has higher coefficient values. The quantization **Table 1** is noted as high quality quantization table and quantization **Table 2** is noted as low quality quantization table in accordance with the properties of quantization effect. The quantization tables for low quality and high quality are tabulated in **Table 1** and **2**, respectively.

Table 1 represents the quality table for quantization of black pixels and **Table 2** represents the Quality table for quantization of white pixels. Higher value in quantization table leads to lower quality of the recovered image and vice versa. The PSNR values of different pairs of Q1 and Q2 are compared in **Table 3**.

The image shown in **Fig. 5** indicates the source image which may be a gray scale image or color image, which is to be encoded and compressed by quality factors QT1 and QT2. **Fig. 6** represents the binary images or black and white images, which is to be embedded in to source image and **Fig. 7** represents the compressed and encoded RGB image. In the above process, the size of original image is 768 Kb and the size of the compressed and encoded image is 262 Kb. Hence it provides a compression ratio of 34.1%. The Encoded Quality factor may vary from 100 to 70 and leads to degradation in quality of the recovered image.

5. DISCUSSION

The performance of the proposed algorithm is analyzed in terms of PSNR, which is calculated by comparing the host image with embedded image. The experimental results show that, the proposed compression technique efficiently compresses and encodes the input images.

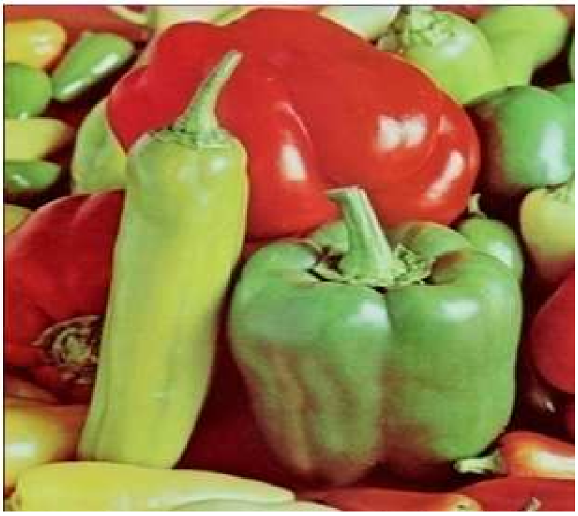

Fig. 5. Original Source (host) Image

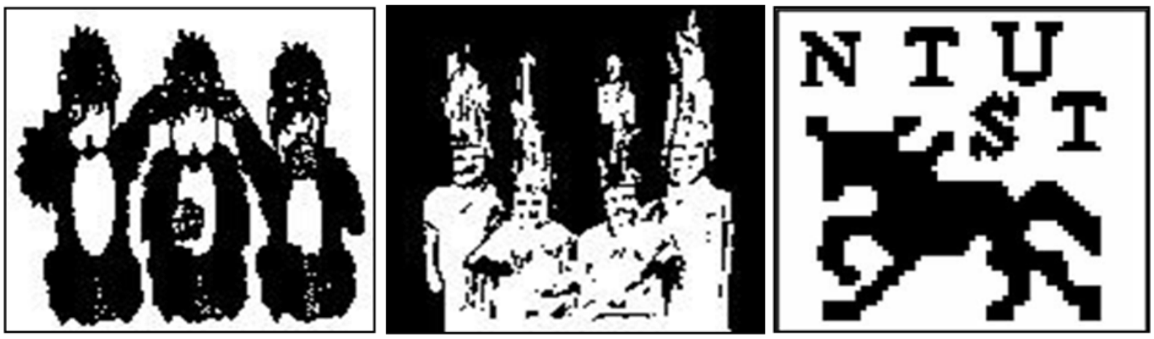

Fig. 6. Gray scale secret images to be embedded

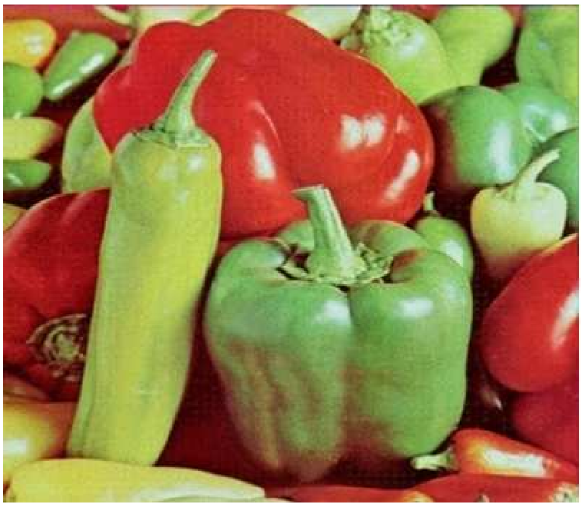

Fig. 7. Compressed and encoded image

Table 3. Comparison of PSNR (dB) for various pairs of Q1 and Q2

Decoded quality factor	Encoded quality factor						
	100	95	90	85	80	75	70
100	36.87	35.60	34.20	33.00	32.00	31.00	28.10
95	35.16	34.10	33.00	32.00	31.80	30.72	27.40
90	34.12	33.98	32.35	31.29	30.87	29.16	28.81
85	33.16	32.10	32.30	30.00	29.40	29.80	29.60
80	32.12	30.90	29.80	28.98	27.18	27.10	26.80
75	31.89	30.12	29.37	28.19	27.36	26.19	25.19
70	29.10	28.19	27.00	25.17	27.30	24.10	22.12

Table 4. Comparison of PSNR (dB) values of various images

PSNR	10%	20%	30%	40%
Lena.jpg	76	65	61	45
Barabara.jpg	75	67	56	51
Baboon.jpg	78	66	46	45
Kotak.jpg	73	67	62	32
Cameraman.jpg	72	62	58	57

Table 5. Performance comparison of PSNR (dB) values with existing methods

	Methodology		
Parameter	Proposed method	Mulla *et al*. (2013)	Seeli and Jeyakuma (2012)
PSNR	76	34.66	35.26

The values of PSNR for various images at different noise levels are being tabulated in **Table 4** of Results section. The performance of the proposed method is compared in terms of PSNR with various existing methods. The comparison is shown in **Table 5**.

6. CONCLUSION

In this study, we have developed a novel scheme for image security based on hybrid compression techniques. The proposed algorithm is based on different quantization tables and produces different image quality. The important function of this scheme is based on quantization table and encryption key. The hybrid compression scheme has shown significantly better results compared with the other compression schemes. The main limitation in this study is that high latency.In future, this technique can be further extended to encrypt and compress the real time videos.

7. REFERENCES

Celik, M.U., G. Sharma, A.M. Tekalp and E. Saber, 2005. Lossless generalized-LSB data embedding. IEEE Trans. Image Process., 14: 253-266. DOI: 10.1109/TIP.2004.840686

Iwamoto, M., 2012. A weak security notion for visual secret sharing schemes. IEEE Trans. Inform. Forens. Security, 7: 372-382. DOI: 10.1109/TIFS.2011.2170975

Johnson, M., P. Ishwar, V.M. Prabhakaran, D. Schonberg and K. Ramchandran, 2004. On compressing encrypted data. IEEE Trans. Signal Process., 52: 2992-3006. DOI: 10.1109/TSP.2004.833860

Kumar, A.A. and A. Makur, 2009. Lossy compression of encrypted image by compressing sensing technique. Proceedings of IEEE Region 10th Conference TENCON, Jan. 23-26, Singapore, pp: 1-5. DOI: 10.1109/TENCON.2009.5395999

Lian, S., Z. Liu, Z. Ren and H. Wang, 2007. Commutative encryption and watermarking in video compression. IEEE Trans. Circuits Syst. Video Technol., 17: 774-778. DOI: 10.1109/TCSVT.2007.896635

Memon, N. and P.W. Wong, 2001. A buyer-seller watermarking protocol. IEEE Trans. Image Process., 10: 643-649. DOI: 10.1109/83.913598

Mulla, O., F. Agada, D. Dawson and S. Sood, 2013. Deep lobe parotid pleomorphic adenoma presenting as obstructive sleep apnoea. BMJ Case Rep. DOI: 10.1136/bcr-2013-008655

Seeli, D.S. and M.K.A. Jeyakuma, 2012 Study on Fractal Image Compression using Soft Computing Techniques. Int. J. Comput. Sci., 9: 420-430.

Wang, R.Z., 2009. Region incrementing visual cryptography. IEEE Signal Process. Lett., 16: 659-662. DOI: 10.1109/LSP.2009.2021334

Zhou, Z., G.R. Arce and G. Di Crescenzo, 2006. Half tone visual cryptography. IEEE Trans. Image Process., 15: 2441-2453. DOI: 10.1109/TIP.2006.875249