JPEG Encryption with File Format Preservation and File Size Reduction

Suah Kim
Department of Information Security
Korea University
Seoul, South Korea
suahnkim@gmail.com

Hyoung Joong Kim
Department of Information Security
Korea University
Seoul, South Korea
khj-@korea.ac.kr

***Abstract*—We present a novel format preserving JPEG encryption scheme, which also losslessly reduces the files size by compressing the quantized DC coefficients. The main idea behind this paper is to introduce additional lossless compression as a part of format preserving encryption to reduce the redundancy as well as encryption. The result reduces the encrypted file size up to 3.2%.**

***Index Terms*—JPEG, Encryption, File size preservation, File format preservation, FPE, File size reduction**

I. Introduction

In the field of encryption, the assumption is that the source of the information is compressed as much as possible before the encryption is applied. Thus, it is generally accepted that the compression should not be part of the encryption algorithm. However, in the context of format preserving encryption, an encryption scheme which is useful for interoperability and data analytic over encrypted domain, this assumption becomes invalid due to the possibility that the format which the source is stored in may not be as compressed as it could be. JPEG is one such format. Due to requirements of low computation and space complexity, JPEG compressed images do not represent the most compressed version of the source image. By offering a compressive feature in the encryption without breaking the file format, we achieve file format preserving JPEG encryption which also reduces the file size.

Past format preserving encryption (FPE) techniques are based on encryption of the quantized DCT coefficients [1]–[5], because they represent the compressed visual information, which makes them ideal for encryption to make the image unrecognizable. However, most of the existing work result in encrypted image file with increase file size or they are not file format preserving.

Niu *et al.* [6] proposed a format preserving JPEG encryption which do not significantly increase the file size. Later on, Shreyamsha Kumar *et al.* [7] proposed a scheme, which also do not significantly result in increased file size. Finally, the most recent work by He *et al.* [8], improves upon existing work by suggesting extended criteria to consider for file format compatibility when using encryption. They also compare the advantages and disadvantages of different encryption methods which ensure file size preservation.

The proposed method improves the idea of the file size preservation by offering a file size reduction as a feature to file format preserving JPEG encryption.

II. Proposed method

To improve the readability, quantized DCT coefficients, quantized DC coefficient, and quantized AC coefficients are denoted as **DCT**, **DC**, and **AC**, respectively, from hereon.

The proposed encryption includes an improved prediction of **DC**, which replaces the DPCM, to increase the overall compression rate. To ensure that the encryption is file format preserving, the prediction is designed to be file format preserving as well.

A. File format preserving quantized DC prediction

The proposed method takes advantage of the DPCM in JPEG, where **DC** is not encoded directly but only the difference between it and the neighbor is encoded. Instead of using the neighbor, we propose making a simple prediction and then use the difference between the original and the predicted value.

For the prediction, we first notice that **DC** value of pixel block (which is denoted as $\text{Block}_{\text{Pixel}}$) is:

$$\mathbf{DC} = \left[\frac{8}{QF_{DC}} \times (mean(\text{Block}_{\text{Pixel}}) - 128) \right] \quad (1)$$

where QF_{DC} represents the quantization entry used for quantizing the DC value, and [.] represents the rounding function.

Thus, a good way to predict the **DC** is by predicting the mean value of the block, using all the neighboring pixels which are already decoded. Fig. 1 shows the blocks used for the proposed prediction: "T", represents that target block that we would like to predict the **DC** of, "N" represents the block which is north from T, "W" represents the block which is West from T. The gray pixels represent the already decoded pixel values that are closest to T.

Since the gray pixels are closest to T, the mean value of the gray pixels can be used to predict the mean value of T:

$$mean(T) \approx \frac{N_{56} + ... + N_{64} + W_8 + ... + W_{64}}{16} \quad (2)$$

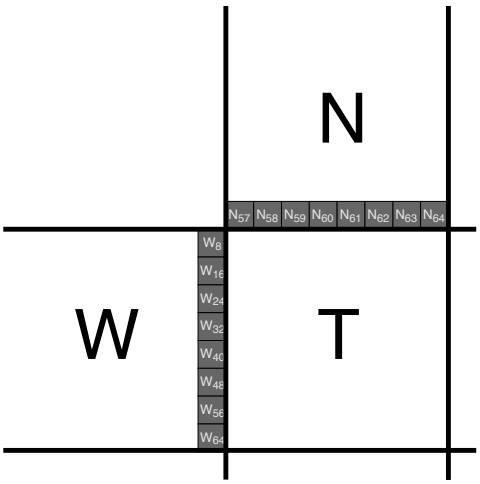

Fig. 1: Context used to predict **DC** of block T. Already decoded neighboring pixels are used to predict.

Substituting approximated $mean(T)$ to Eq.1, we can estimate **DC**, which is denoted as $\hat{\mathbf{DC}}$:

$$\hat{\mathbf{DC}} = \left[\frac{8}{QF_{DC}} \times \left(\frac{N_{56} + ... + N_{64} + W_8 + ... + W_{64}}{16} - 128 \right) \right] \quad (3)$$

B. Encryption

Once the prediction is finished, the prediction error of **DC** value ($\mathbf{\Delta DC} = \mathbf{DC} - \hat{\mathbf{DC}}$) and **AC** values are encoded using Huffman coding and run-length coding then, they are encrypted using the a file size preserving JPEG encryption (permutation and symbol-wise encryption). For encoding the **DC** values, DPCM is not used and $\mathbf{\Delta DC}$ are directly encoded. The original **DC** values can be recovered during decryption by making the same prediction for **DC** and adding the decrypted prediction errors: *i.e.,* $\mathbf{DC} = \hat{\mathbf{DC}} + \mathbf{\Delta DC}$.

Although the size of the **AC** will not change significantly, the size of $\mathbf{\Delta DC}$ is likely to be smaller, thus the sum of two will be smaller than the original JPEG file size.

III. Experimental Results

For the experiment, we use the six of the most well known images from USC-SIPI image database: Lena, Boat, Barbara, Baboon, Peppers, and F16. Each image is first compressed using standard JPEG and then encrypted with the proposed method.

Fig. 2 shows that the proposed method achieves higher file size reduction for lowest QF (file size reduction of 3.2% for image Lena at $QF = 30$). The lowest file size reduction overall occurs for image Baboon. This is not a surprising result because image Baboon has many high frequency components, making it hard to compress.

IV. Conclusion

The proposed method proposes a file format preserving JPEG encryption with lossless file size reduction capability up to 3.2%. The method takes advantage of the JPEG's inefficiency in compressing the quantized DC. By carefully integrating the quantized DC prediction scheme in the file format preserving encryption, the file size of the encrypted is reduced while preserving the file format.

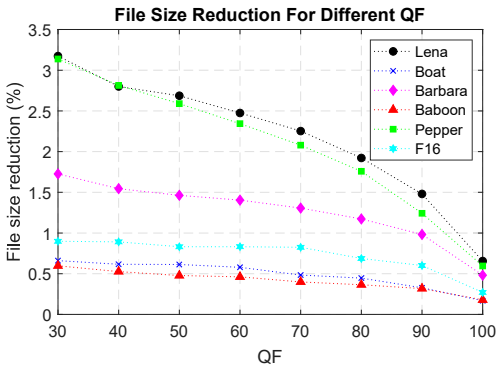

Fig. 2: File size reduction rate for different QF factors.

V. Acknowledgment

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (No.NRF-2019R1I1A1A01059582) and partly supported under the framework of international cooperation program managed by the National Research Foundation of Korea (2018K2A9A2A06024168, FY2019).

References

[1] K. Minemura, Z. Moayed, K. Wong, X. Qi, and K. Tanaka, "Jpeg image scrambling without expansion in bitstream size," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2013, pp. 261–264.

[2] A. Unterweger and A. Uhl, "Length-preserving bit-stream-based jpeg encryption," in *Proceedings of the on Multimedia and security*. ACM, 2012, pp. 85–90.

[3] S. Y. Ong, K. Minemura, and K. S. Wong, "Progressive quality degradation in jpeg compressed image using dc block orientation with rewritable data embedding functionality," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 4574–4578.

[4] S. Auer, A. Bliem, D. Engel, A. Uhl, and A. Unterweger, "Bitstream-based jpeg encryption in real-time," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 5, no. 3, pp. 1–14, 2013.

[5] B. Kishore, B. S. Kumar, and C. R. Patil, "Fpga based simple and fast jpeg encryptor," *Journal of Real-Time Image Processing*, vol. 10, no. 3, pp. 551–559, 2015.

[6] X. Niu, C. Zhou, J. Ding, and B. Yang, "Jpeg encryption with file size preservation," in *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. IEEE, 2008, pp. 308–311.

[7] B. Shreyamsha Kumar and C. R. Patil, "Jpeg image encryption using fuzzy pn sequences," *Signal, image and video processing*, vol. 4, no. 4, pp. 419–427, 2010.

[8] J. He, S. Huang, S. Tang, and J. Huang, "Jpeg image encryption with improved format compatibility and file size preservation," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2645–2658, 2018.