Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme

[a]Jiri Fridrich, [b]Miroslav Goljan, and [c]Nasir Memon

[a]Department of Systems Science and Industrial Engineering, SUNY Binghamton, Binghamton, NY
[b]Department of Electrical Engineering, SUNY Binghamton, Binghamton, NY
[c]Department of Computer Science, Polytechnic University, Brooklyn, NY

ABSTRACT

In this paper, we describe new and improved attacks on the authentication scheme previously proposed by Yeung and Mintzer [1]. Previous attacks assumed that the binary watermark logo inserted in an image for the purposes of authentication was known. Here we remove that assumption and show how the scheme is still vulnerable, even if the binary logo is not known but the attacker has access to multiple images that have been watermarked with the same secret key and contain the same (but unknown) logo. We present two attacks. The first attack infers the secret watermark insertion function and the binary logo, given multiple images authenticated with the same key and containing the same logo. We show that a very good approximation to the logo and watermark insertion function can be constructed using as few as two images. With colo images, one needs many more images., nevertheless the attack is still feasible. The second attack we present, which we call the `collage-attack' is a variation of the Holliman-Memon counterfeiting attack. The proposed variation does not require knowledge of the watermark logo and produces counterfeits of superior quality by means of a suitable dithering process that we develop.

1. INTRODUCTION

Yeung and Mintzer [1] proposed the following method for authentication of digital images. The process of image authentication starts with a secret key that is used to generate a key dependent binary valued function f, f: $\{0, 1, \ldots, 255\} \rightarrow \{0,1\}$, that maps integers from 0 to 255 to either 1 or 0. For color images, three such functions (or binary lookup tables), f_R, f_G, f_B, one for each color channel, are generated. These binary functions are used to encode a binary logo L. In order to embed a watermark, gray scale images are perturbed to satisfy the following expression

$$L(i,j) = f_g(g(i,j)) \text{ for each pixel } (i,j). \tag{1}$$

For an RGB image, the three color channels are perturbed to obtain

$$L(i,j) = f_R(R(i,j)) \oplus f_G(G(i,j)) \oplus f_B(B(i,j)) \text{ for each pixel } (i,j). \tag{2}$$

In addition to this process, since original pixel values are modified during watermark insertion, Yeung and Mintzer use error diffusion to maintain proper average color over the image in any local region. Although the error diffusion step is crucial in suppressing any annoying artifacts that might be introduced during watermark insertion, it is not of interest in our discussion. This is because, for all practical purposes, the image that is obtained after watermark insertion is treated as ``the'' original image whose integrity is to be verified by subsequent extraction of the embedded watermark logo and checking it for an modifications. Hence any changes made while arriving at this “original” image are not of interest to a potential attacker. Furthermore, in later versions of the technique, Yeung and Mintzer also scramble the binary logo prior to embedding. Again this scrambling process is not relevant to the type of attacks described here and hence for the sake of brevity we chose to ignore it.

Once a watermark is inserted, image authenticity is then easily verified by checking the relationship $L(i,j) = f_g(g(i,j))$ for each pixel (i,j). Note that the authentication process requires knowledge of the binary lookup tables or the secret key used to generate them.

There are some obvious advantages of this approach. First, the logo itself can carry some useful visual information about th image or its creator. It can also represent a particular authentication device or software. Second, by comparing the original logo with the recovered one, one can visually inspect the integrity of the image and readily identify cropping. Third, the authentication watermark is embedded not only in the LSB's of the image but somewhat deeper ± 5 gray scales). This makes it somewhat more secure and harder to remove. Fourth, the method is fast, simple, and amenable to fast and cheap hardware implementation. This makes it very appealing for still image authentication in digital cameras or for authentication of digital video.

However, in order to use this authentication scheme in digital cameras, one needs to clarify the important issue of key (and logo) management. To the best knowledge of the authors, this issue has not been properly addressed in the literature before. In Section 2, we demonstrate that if the same logo and key are reused for several images, one could easily reconstruct the logo as well as the binary lookup tables from a relatively small number of images. One possibility how to make this attack more difficult to mount is to let the binary functions depend not only on the colors but also on the pixel positions i, j as suggested in [5]. That this measure is not enough is demonstrated in Section 3 in which we present another counterfeiting attack (the collage attack) and optimize its performance. The collage attack was introduced first by Holliman and Memon in [2] as a special case of a substitution or counterfeiting attack, which applied to a large class of block-based watermarking techniques. However, Holliman and Memon assumed the attacker knows the binary logo. Here we demonstrate that the attack can still be carried out even if the logo is not known. Furthermore, we develop special dithering techniques to improve the quality of the counterfeit that is constructed. We demonstrate how the quality of counterfeit images (an aspect ignored b Holliman and Memon) can be significantly improved with the proposed dithering process. Finally, in Section 4 we conclude the paper and discuss simple modifications that can be made to the Yeung-Mintzer scheme to prevent the kind of attacks described here.

2. INFERRING THE SECRET BINARY FUNCTION AND THE WATERMARK LOGO

Recently, Memon, Shende and Wong [5] showed that the uncertainty in the secret binary functions or lookup tables $f_R()$, $f_G()$ and $f_B()$ can be significantly reduced given a sequence of watermarked images containing the same logo, which was assumed to be known. In this section we show that knowledge of the logo is not necessary. First we show that for gray scale images the look-up tables can be mostly reconstructed using as few as two images. For the color case we need many more images or the watermark image needs to have sufficient "structure".

Case 1: Gray scale images - If the same logo lookup tables are used for multiple images, both the logo and the binary functions can be almost completely recovered from as few as two grayscale images. Given two watermarked grayscale $M\times N$ images U and V with gray scales u and v, watermarked with the same secret ke K and a binary logo B, one can write

$$L(i,j) = f_g(u(i,j)) = f_g(v(i,j)) \text{ for every pixel } (i,j). \tag{3}$$

This constitutes $M\times N$ equations for 256 unknowns $f_g(0)$, $f_g(1)$, …, $f_g(255)$. Most of the equations are redundant and, depending on the image, there may not be a unique solution f_g. We can start with the set {0, 1, .., 255} divided into 256 subsets, each subset having exactly one gray scale level. Then, the first equation, $f_g(u(1,1)) = f_g(v(1,1))$, tells us that the values of f_g are the same for both $u(1,1)$ and $v(1,1)$. Thus, we can group together these two gray levels because the value of the binary function f_g is the same for both. Gradually, the 256 subsets will start to coalesce into a smaller number of larger subsets. At the end, there will be two large subsets, one corresponding to $f_g = 0$, the other to $f_g = 1$, and several other sets for which the value of f_g is undetermined.

To test the plausibility of this idea, we have performed experiments with several grayscale images. Two test images 256×256 have been watermarked with the same binary logo and the same binary function f_g. The system of equations (3) was then solved to recover the binary function f_g. Three test images, 'Lena', 'Airfield', and 'Bridge', are shown in Figures 1−3. Using the images 'Lena' and 'Airfield', we were able to determine 243 values for f_g. Combining the watermarked images 'Airfield' and 'Bridge' gave us 232 values for f_g. This is an unacceptable leak of information, which clearly indicates that the binary lookup table and the logo should not be reused for more than one image. With increasing number of available images, the number of correctly recovered values of the binary function f_g quickly saturates at 256.

Figure 1 Test image 'Lena'

Figure 2 Test image 'Airfield

Figure 3 Test image 'Bridge'

Case 2: Color images : In [5], Memon, Shende and Wong showed that if the binary logo L is known than the uncertainty in the binary function f_R, f_G, and f_B can be significantly reduced. They concluded that, in some cases, the binary look-up tables can be reduced to as few as four possibilities, given a single image. However, as we show here, knowledge of the binary logo L is really not necessary. The binary functions f_R, f_G, and f_B can still be mostly reconstructed provided we have a sufficient number of images containing the same logo and watermarked using the same key. That is, a similar attack as th gray scale case described above can also be mounted for color images, although the number of images needed for reconstructing the binary functions f_R, f_G, and f_B is much higher. This is because whereas in the grayscale image case we had 256 sets to begin with, in the color case we have as many as 2^{24} for 24 bit images. Assuming a 24 bit color image for the sake of discussion, we start with the colors $(0, \ldots, 2^{24}-1)$ partitioned into 2^{24} sets, with each color being in a separate set. Then, given a sequence of images $X_1, \ldots, X_k$, let $X^{i,j}$ denote the set of pixels in the sequence at location (i,j). If the same logo is embedded in each image in the sequence $X_1, \ldots, X_k$ then all the pixels in the set $X^{i,j}$ map to the same value $L(i,j)$, which can be either 0 or 1. Now if two such sets say $X^{i,j}$ and $X^{k,l}$ have a non-empty intersection, then clearly all pixels in these two sets again map to the same value and hence can be grouped together. We continue in this manner grouping together sets that have a non-empty intersection and stop when we have all pixels partitioned into exactly two sets. In practice we do not expect to get exactly two sets but if most of the pixels fall into two sets then we have significant information about the binary lookup tables. That is, we know the value of $f_R(\cdot) \oplus f_G(\cdot) \oplus f_B(\cdot)$ for a large majority of RGB triples. Note that this would be sufficient to make modifications to an image and still have it authenticate. However we still do not know the individual functions f_R, f_G, and f_B. These can be deduced by using the techniques given in [5] or by directly solving a system of $M \times N$ linear equations in mod 2 arithmetic. No formal analysis of this issue has been performed as yet. The authors believe that the fact that th watermarking technique has certain serious security weaknesses for gray-scale images is a sufficient argument fo modification

Note that replacing the linear expression (2) with a more complicated expression involving quadratic or other non-linearity's with the hope to make the numerical solution of the system of equations impractical, is not likely to bring a significant improvement. This is because the number of equations exceeds the number of unknowns by a large margin and the values of the nonlinear terms can simply be added as another set of unknowns. In Section 4, we propose better countermeasures agains this attack.

3. THE COLLAGE ATTACK

The second problem described in this paper is actually common to all watermarking schemes in which the watermark is embedded locally and does not depend on the image index. This attack will apply to even more general schemes in which th binary function depends not only on the gray level (color) but also on the pixel position i, j in the image. We call the attack the *collage attack*. The collage-attack is a variation of the Holliman-Memon counterfeiting attack on blockwise independent watermarking techniques [2]. However, in their attack of the Yeung-Mintzer technique, Holliman and Memon had assumed the watermark logo is available to the attacker. In this paper, we show that the attack is possible even when the binary logo is not known. In this case the attacker just needs a larger number of images watermarked with the same key and logo. Furthermore, Holliman and Memon did not consider the quality of the reconstructed image in their attack. Here we show that unless care is taken the counterfeit image could be of poor quality. We then give techniques for producing high qualit counterfeit images.

The basic idea behind the collage attack is to create a new image from multiple authenticated images by combining portions of different images while preserving their relative spatial location within the image. More generally, given multiple $M\times N$ images I_1, I_2, …, I_n authenticated with the same logo and binary functions, we can construct a new image (a collage) J by selecting pixels from all n images, while preserving the relative location of the pixels in the images:

$$J(i,j) = I_k(i,j) \text{ for some } k\in\{1, \ldots, n\} \text{ and all } (i,j).$$

The new image J will be authentic if its authenticity is checked with the same logo and binary functions used fo authenticating images I_1, …, I_n. Given a large number of images I_1, I_2, …, I_n with RGB colors $(R_k(i,j), G_k(i,j), B_k(i,j))$, $k = 1, \ldots, n$, a pirate can take an arbitrary image I with RGB colors $(R(i,j), G(i,j), B(i,j))$, and create an approximation to I by selecting the closest colors from all n images for every pixel

$$(R, G, B) = (R_k, G_k, B_k) \text{ with } k = \min \arg_k \{ (R-R_k)^2 + (G-G_k)^2 + (B-B_k)^2 \}, \tag{4}$$

where the last expression needs to be evaluated for every pixel (i,j). In the next sub-section we calculate the expected value of the difference $\| I - J \|$ (in square norm) depending on the number of images in the database, n, based on some simple assumptions about the images. We then introduce a modified dithering algorithm to further improve the visual quality of the forgery J. The section is closed with several examples of forgeries generated from a database containing n = 20, 80, and 300 images.

Calculation of Expected Distortion in Counterfeit: We model the statistical distribution of colors in the images simply as a uniform distribution of colors in the color cube. In what follows, we also assume that the space of all possible RGB colors is scaled to a unit cube C and that the color values are triples of real numbers. Let D be the distance between two randoml chosen colors in the color cube C. The function $F(x) = P(D < x)$ is a non-decreasing probability distribution function with $F(0) = 0$, $F(a) = 1$ ($a = \sqrt{3}$ for Euclidean metric in the color cube and $a = 1$ for the maximal metric). We further denote $F_{\geq}(x) = P(D \geq x)$. The expected value $E(D)$ of the distance between two randomly chosen colors in C is

$$E(D) = \int_0^a x.F'(x)dx = \int_0^a F_{\geq}(x)\, dx\,.$$

We approximate the average error due to quantization in Eq. (4) as the expected value of the distance between a randoml chosen color and the nearest color out of other n randomly chosen colors

$$E(D,n) = \int_0^a F_{\geq}^n(x)dx\,.$$

The probabilit $F_{\geq}(x) = P(D \geq x) = \int_{z\in C} V(B(z,x)\cap C)\, dz = 1 - 4/3\, \pi x^3 + O(x^4)$, where $V(\,)$ denotes the volume and $B(z, x)$ is a ball with center in z and radius x. To find out about asymptotic behavior of this integral for large n, it will become clear that it is sufficient to calculate the cubic term only. Now we need to find the asymptotic expression fo $E(D, n)$ for large n:

$$E(D,n) = \int_0^a F_{\geq}^n(x)dx \approx \int_0^a (1-\frac{4}{3}\pi x^3 + O(x^4))^n dx\,.$$

The asymptotic behavior of this integral for large n can be obtained using the well-known Laplace technique [4, pp. 36]

$$E(D,n) \approx \frac{1}{3}\Gamma(\frac{1}{3})(\frac{3}{4\pi n})^{1/3} = \frac{1}{6}\Gamma(\frac{1}{3})\sqrt[3]{\frac{3}{\pi}}\sqrt[3]{\frac{1}{n}} = 0.554\sqrt[3]{\frac{1}{n}}\,.$$

Finally, the average mean square error between the original image I and its forgery J obtained from a database of n images is

$$\| I - J \| = \frac{1}{MN} \sum_{i,j} [I(i,j) - J(i,j)]^2 \approx 0.3073\sqrt{\frac{1}{n^2}} \, .$$

Improving Counterfeit Quality by Error Diffusion: It is possible to improve the visual appearance of the forgery b diffusing the quantization error using a mechanism similar to error diffusion. However, the standard error diffusion does not work well in this case because the palette is different for every pixel and the palette color distribution does not fill out the color cube uniformly. The resulting effect is that error diffusion mechanism causes a "spillage" of colors when the diffusion comes to a boundary between two colors. This leads to poor forgeries that actually look worse than the simple approximation based on Eq. (4). To fix this problem, we employ a decay factor that suppresses the error and avoids the above-mentioned spillage of colors. The new colo c_i' at pixel i can be expressed using the original colo c_i as

$$c'_i = c_i - decay \times erro_{\ i}$$
$$erro_{\ i} = |\, closest\_color_{i-1} - c'_{i-1} \,| \, .$$

We note that decay=1 in the classical error diffusion technique. The purpose of the decay factor is to prevent the error from growing due to a lack of colors at a given pixel. The error grows especially fast in regions containing a uniform color c while the available colors from the image database are all far from that colo c. This happens when the colo c is located close to a face, an edge or the vertex of the color cube. Let us look at the case of the vertex first. Let D_v be the distance of a randoml chosen color from a vertex. We have

$$P(D_v \geq x) = (1 - 4/3\pi x^3/8) \, , \text{ for } x \leq 1.$$

The expression for the probability becomes more complicated fo $x > 1$ but, fortunately, it is enough to notice that

$$max(P(D_v \geq x)) = m < 1 \text{ for } 1 < x \leq \sqrt{3}$$

to obtain

$$E(D_v, n) = \int_0^1 [1 - \frac{4}{3}\pi(x/2)^3]^n dx + O(m^n) \, .$$

Therefore

$$E(D_v, n) \approx \int_0^1 [1 - \frac{4}{3}\pi(x/2)^3]^n dx = 2\int_0^{1/2} [1 - \frac{4}{3}\pi t^3]^n dt = 2E(D,n) \, .$$

The expected values for an “edge” and “face” colors can be calculated similarly:

$$E(D_e, n) \approx \int_0^1 [1 - \frac{1}{4}\frac{4}{3}\pi x^3]^n dx = \sqrt[3]{4}\int_0^{1/\sqrt[3]{4}} [1 - \frac{4}{3}\pi t^3]^n dt = \sqrt[3]{4}E(D,n) \, ,$$
$$E(D_f, n) \approx \int_0^1 [1 - \frac{1}{2}\frac{4}{3}\pi x^3]^n dx = \sqrt[3]{2}\int_0^{1/\sqrt[3]{2}} [1 - \frac{4}{3}\pi t^3]^n dt = \sqrt[3]{2}E(D,n) \, .$$

The accumulated error after processing K pixels in a row becomes

$$erro_{\ i+K} \leq decay \times (\, erro_{\ i+K-1} + decay \times (\, erro_{\ i+K-2} + \ldots + decay \times error_i)) <$$
$$< decay \times erro_{\ i+K-1} + decay^2 \times \, erro_{\ i+K-2} + \ldots$$

Because $E(erro_{\ i+K}) \leq E(D_v, n)$, we have

$$E(erro_{\ i+K}) \leq \, decay/(1 - decay) \times E(D_v, n) \text{ for all } K.$$

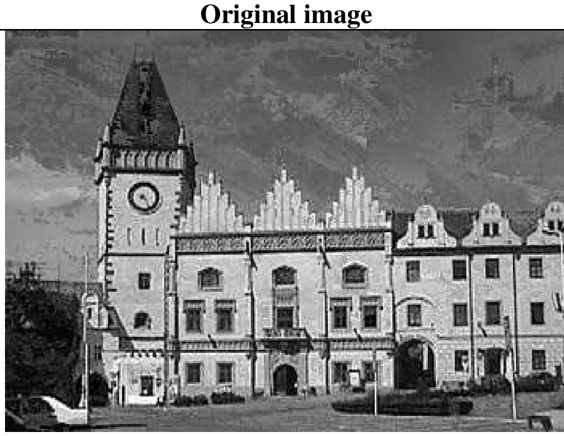
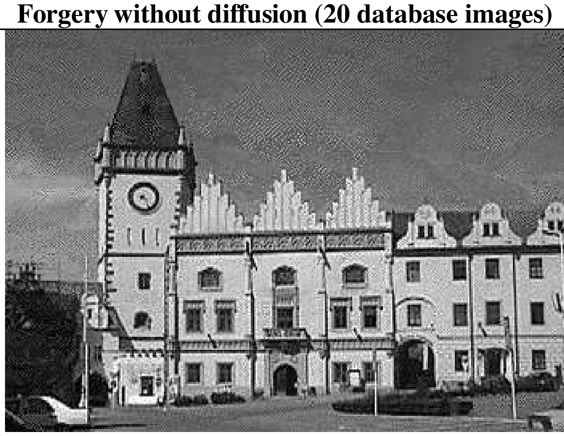
 Original image	 Original image
 Forgery without diffusion (20 database images)	 Forgery without diffusion (20 database images)
 Forgery with diffusion (20 database images)	 Forgery with diffusion (20 database images)

Figure 4 Examples of two images and forgeries obtained using the simple approach based on Eq. 4 (without error diffusion) and with the diffusion and the optimal decay factor (with diffusion). Only 20 database images were used i this Figure.

This error should not grow above certain maximal value during our dithering process. In particular, we do not want to obtain a color outside the color cube. Thus, we require that the error be smaller than one half of the diameter of the cluster of available colors [$\sqrt{}3-2\,E(D_v, n)$]/2. This gives us an upper bound fo *decay*:

$$decay/(1-decay)\times E(D_v, n) \leq [\sqrt{}3-2\,E(D_v, n)]/2, \text{ or}$$
$$decay \leq 1-2/\sqrt{}3E(D_v, n).$$

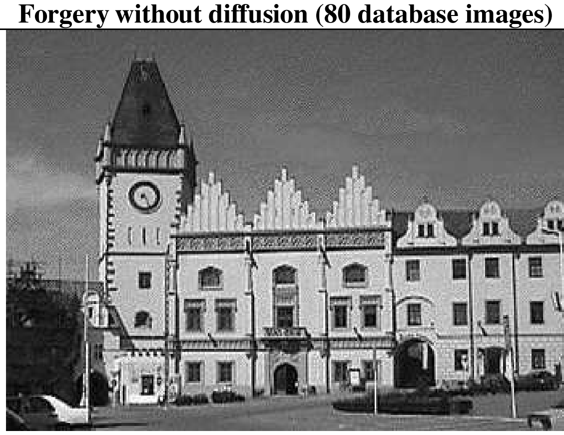
On the other hand, the *decay* should not be too small otherwise no error would be diffused to neighboring pixels. Consequently, we choose the decay value equal to the upper bound

$$decay = 1-2/\sqrt{}3E(D_v, n) = 1-\frac{4\beta}{\sqrt[2]{3}\sqrt[3]{n}}\text{, where } \beta=\frac{1}{6}\Gamma(\frac{1}{3})\sqrt[3]{\frac{6}{\pi}}\cong 0.554\,.$$

Our numerical experiments with different images and decay factors indeed confirmed that this value of decay gives us the most pleasing results (see Figure 4, 5, and 6). We have performed a series of experiments on a database consisting of 300 color images with 350×250 pixels. As these figures show, the improvement obtained by the proposed diffusion process is indeed significant, especially when fewer images are available for constructing a forgery.

Forgery without diffusion (80 database images)

Forgery without diffusion (80 database images)



Forgery with diffusion (80 database images)

Forgery with diffusion (80 database images)

Figure 5 Examples of two images and forgeries obtained using the simple approach based on Eq. 4 (without error diffusion) and with the diffusion and the optimal decay factor (with diffusion). Total of 80 database images were used in this Figure.

Original image

Forgery with diffusion (300 database images)

Original image

Forgery with diffusion (300 database images)

Original image

Forgery with diffusion (300 database images)

Figure 6 Examples of images and their forgeries obtained using a database of 300 images and error diffusion with a decay factor.

4. CONCLUSION AND COUNTERMEASURES

There are several possibilities how to prevent the attacks described in Sections 2 and 3. In [5] Memon, Shende and Wong suggest making the watermark extraction function dependent on the position of the pixel This can be done, for example by adding two more look-up tables f_I and f_J that take the row index i and column index j and also output a zero or a one value which are then XOR'ed with the values produced by the f_R, f_G and f_B look-up tables to yield the watermark bit. In other words, we have

$$L(i,j) = f_R(R(i,j)) \oplus f_G\{G(i,j)) \oplus f_B(B(i,j)) \oplus f_I(i) \oplus f_J(j).$$

Although the above strategy makes the attacks outlined in section 2 ineffective, it is not enough to prevent the collage attacks presented in Section 3.

It appears that the simplest solution would be to make the watermark depend on an image index. This would solve the collage attack completely, and since the binary functions f and the logo L can also be made index-dependent, we could not apply the first attack either. However, now we need the index in order to authenticate the image. An exhaustive search may not be a plausible approach because of a potentially large number of authenticated images (video-frames, for example). A better idea would be to embed the index in the image using a key uniquely associated with a particular digital camera or a movie.

The index cannot be embedded in the whole image because tampering with a portion of the image would result in a conclusion that the whole image has been tampered with. Thus, the index needs to be embedded multiple times in small blocks. The index, however, should be embedded in a robust manner so that the correct index is extracted even from slightl modified blocks. However, robust embedding leads to larger distortion contributing to the distortion due to the fragile watermark. Obviously, the robust watermark must be embedded as the first watermark. To achieve at least moderate robustness of the index watermark, we may have to increase the block size to at least 64×64 pixels. Assuming the index can be captured using 32 bits, we are taking a potentially large risk that tampering with a very spatially localized block feature will cause an erroneous index extraction. Increasing the block size will negatively influence the ability of the authentication scheme to localize changes.

Hence we propose a different approach to thwart the attacks described in Sections 2 and 3. We observe that making the binary function f depend on more than one pixel can thwart the first attack. For example, assuming we are scanning the picture by rows, we can include a fixed random collection of neighbors from the portion of the image that has already been modified. The lookup tables can be replaced with a mapping derived from a secure block encryption algorithm to prevent attacks based on calculating the lookup tables. The attack described in Section 2 will not be successful for this scheme because of a large number of possible combinations of values of the binary function f. It is also not known which pixels contribute to the current pixel. On top of this, the secure block encryption algorithm provides additional feature of securit making the first attack impractical.

To enable detection of collages, we need to embed the image index into the image. We suggest to embed this index into every small, 32×32 block into randomly chosen pixels (the selection of pixels is the same for every block). The embedding of the index can be integrated into the authentication scanning process described in the above paragraph simply by making an additional request whenever we are at a pixel that will carry information about the index. We request that not onl $f(p)$ = Logo(p) but also $g(p)$ = index bit for some function g.

In the authentication step, we first apply the detection function for the fragile watermark and by looking at the logo, we can identify portions of the image that have been tampered with. We can also determine if the image has been cropped and identify boundaries of collaged portions from multiple images. To find out if the collaged portions came from different images, the image indices are recovered from each 32×32 block.

ACKNOWLEGEMENTS

The work on this paper was partially supported by Air Force Research Laboratory, Air Force Material Command, USAF, under the grants number F30602-98-C-0176 and F30602-98-C-0009. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions

contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Air Force Research Laboratory, or the U. S. Government.

REFERENCES

[1] M. Yeung, and F. Mintzer. "An Invisible Watermarking Technique for Image Verification", *Proc. ICIP'97*, Santa Barbara, California, 1997.

[2] M. Holliman and N. Memon. “Counterfeiting attacks for block-wise independent watermarking techniques.” *To.appear in IEEE Transactions on Image Processing*, March 2000.

[3] M. Holliman, N. Memon, and M. M. Yeung, "On the Need for Image Dependent Keys for Watermarking *", Proc. Content Security and Data Hiding in Digital Media*, Newark, NJ, May 14, 1999.

[4] A. Erdélyi, *Asymptotic Expansion* , Dover Publications, Inc., New York, 1956, pp. 36.

[5] N. Memon, S. Shende and P. Wong. “On the security of the Yeung-Mintzer Authentication Watermark. *Proc. of the IS & T PICS Symposium*, Savannah, Georgia, March 1999.