CRYPTANALYSIS OF THE YEUNG-MINTZER FRAGILE WATERMARKING TECHNIQUE

Jessica Fridrich[a], Miroslav Goljan[b], Nasir Memon[c]

[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

The recent proliferation of digital multimedia content has raised concerns about authentication mechanisms for multimedia data. A number of authentication techniques based on digital watermarks have been proposed in the literature. In this paper we examine the security of the Yeung-Mintzer authentication watermarking technique and show that it is vulnerable to different types of impersonation and substitution attacks whereby an attacker is able to either create or modify images that would be considered as authentic. We present two attacks. The first attack infers the secret watermark insertion function. This enables an attacker to embed a valid watermark in any image. The attack works without knowledge of the binary watermark inserted in the image, provided the attacker has access to a few images that have been watermarked with the same secret key (insertion function) and contain the same watermark. We show simulation results where the watermark and the watermark insertion function can be mostly re-constructed in a few (1-5) minutes of computation, using as few as two images. 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 sophisticated dithering process.

Keywords: Digital Watermarks, Authentication, Yeung-Mintzer, Fragile Watermark, Substitution Attacks.

1. INTRODUCTION

Authentication techniques provide a means of ensuring the integrity of a message. It should be noted that, authentication, in general, is quite independent of encryption, where the intent is to ensure the secrecy of a given message. Authentication codes are essentially designed to provide assurance that a received message has not been tampered with and a specific source is indeed the originator. This could be achieved with or without secrecy. In fact, in certain applications, secrecy could actually turn out to be an undesirable feature of an authentication technique. The general model under which authentication techniques are studied is shown in Figure1

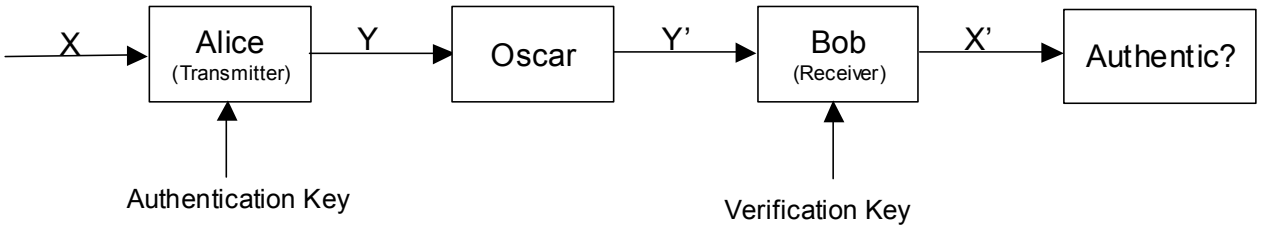


Figure 1: Authentication Model

In this model, we have a transmitter, Alice, and a message *X* that she wishes to transmit to Bob over an open channel. In order for Bob to be assured that the message *X* indeed did originate from Alice, and has not been modified, Alice computes an authenticated message *Y* that she sends over the open channel. *Y* is a function of *X* and a secret authentication key. In general, authentication is achieved by adding redundant information to a message. This redundant message could be in the form of an authentication tag (or authenticator) attached to the end of the message being authenticated. In this case Y would be of the form $Y = (X \,||\, a)$, where *a* is the appended authenticator and || denotes concatenation. Authentication could also be achieved by redundancy present in the structure of the message, which could be recognized by the receiver [1]. Most of the work in authentication assumes the former case.

Now, if Bob receives $Y = (X \,||\, a)$ he could verify, using a verification key, that *a* is indeed a valid authenticator for *X* and accept the message. In a symmetric key system, the authentication and verification key are identical and both need to be kept a secret shared only between Alice and Bob. Since the authenticated message is being transmitted over an open channel, a malicious Oscar, can intercept the message and replace with another message $Y' = (X' \,||\, a')$, which is different from *Y,* and which he hopes Bob would accept as an authentic message. Note that Oscar performs this operation without knowledge of any secret key. Such an attack is called a *substitution attack*.

Oscar may also insert a message *Y'* straight into the channel without knowledge of any authentic message that Alice has sent to Bob. Such an attack is called an *impersonation attack*. Oscar may also choose freely between a substitution attack and an impersonation attack. Authentication techniques that are unconditionally secure against these attacks, from an information theoretic point of view, are known [1]. One problem with the model described above is that Alice can always disclaim originating a message. Non-repudiable authentication techniques are also known that can prevent such a situation. For an excellent recent survey on authentication techniques, the reader is referred to [1].

Closely related to authentication techniques are digital signature schemes and message authentication code (MAC) generation algorithms [2]. The former employs public key techniques to generate a signature for a message that can be verified by anyone having knowledge of the public key (for example, see [3]). Digital signature schemes are usually non-repudiable. MAC techniques are symmetric key (private key) based and in this sense similar to authentication codes (for example, see [4]). However, they only provide computational guarantees about security. That is, generating false messages is known to be (in most cases without any formal proof) computationally intractable. For an excellent introduction to digital signatures and related topics the reader is referred to [2].

The recent proliferation of digital multimedia content has raised concerns about authentication mechanisms for multimedia data. When multimedia content is used for legal purposes, medical applications, news reporting, and commercial transactions, it is important to ensure that the content originated from a specific source and that it was not changed, manipulated or falsified. There have been numerous authentication techniques for multimedia objects proposed in the literature. Most of these techniques appear to have originated in the signal processing literature and are based on digital watermarks. A watermark is a (often secret key dependent) signal added to digital data (namely audio, video or still images), which can later be extracted or detected to make an assertion about the data. In this work, we are concerned with authentication of image data and hence restrict our attention to inserting and extracting watermarks from images. It should be noted however, that the techniques we discuss are quite general and apply equally well to other type of data, including audio and video data.

A digital watermark is said to be *robust* if it is designed to withstand malicious attempts at removal. A user not in possession of the secret key used for watermark insertion is unable to either detect a robust watermark or remove it without causing significant degradation to the original content. On the other hand, a watermark designed to detect any change whatsoever made to the content is known as a *fragile* watermark. For excellent reviews on robust and fragile watermarking techniques, the reader is referred to [5],[6],[7]. Although both fragile [8], [9] and

robust watermarks [10], [11], [12] have been proposed for the purposes of authentication, in this paper we focus on fragile watermarks.

Fragile watermarks can be essentially viewed as a special case of cryptographic authentication techniques. However, one advantage of using fragile watermarks for authentication, as opposed to a conventional authentication technique is that with fragile watermarking, the authenticator is embedded within the content. This greatly simplifies the logistical problem of data handling and incorporating an authentication function in applications that represent images in one of the many possible different data formats that are used in practice today, many of which do not have an explicit support for authentication. Another advantage is that fragile watermarks allow the determination of the exact pixel locations where the image has been modified. Hence, there has been considerable interest in developing fragile watermarks for image data. However, the focus of these efforts has been mainly towards embedding (and extracting) authentication codes in digital signals by means of an appropriate watermark. There has been little attention paid to cryptanalysis of proposed authentication techniques. In this paper we cryptanalyze one such proposed technique, namely the Yeung-Mintzer fragile watermarking technique [14,15], and show that it has a significant weakness and is subject to substitution and impersonation attacks. Although we focus mainly on the Yeung-Mintzer technique, they also apply to some of the attacks we develop are also applicable to other techniques based on the Yeung-Mintzer approach (for example, [17]). Also, we believe that although many fragile watermarks have been proposed in the literature, a thorough cryptanalysis of these techniques has been lacking and this work highlights the importance of conducting a serious investigation of a fragile watermark technique before adopting it for a given application..

The rest of the paper is organized as follows. In the next section we give a brief description of the Yeung-Mintzer watermarking technique. In section 3 we show how an adversary Oscar, can infer the secret watermark insertion function used by the Yeung-Mintzer technique given a few images (just two images appears to be enough in practice) containing the same but unknown watermark logo. In section 4 we investigate another attack that enables us to construct counterfeits or forgeries, given a few images containing the same unknown watermark logo and insertion key. This attack is an improved version of a counterfeiting attack proposed by Holliman and Memon [16]. The proposed variation does not require knowledge of the watermark logo as required by Holliman and Memon, and produces counterfeits of superior quality by means of a sophisticated dithering process. In section 5 we then conclude with a discussion on how the Yeung-Mintzer technique needs to be used and/or modified to make our attacks difficult.

2. THE YEUNG-MINTZER AUTHENTICATION WATERMARK

The Yeung-Mintzer technique [14, 15] is perhaps one of the earliest and most cited fragile watermarking technique in the literature. In this technique, a watermark image *W* (usually a binary logo image) is embedded into a source image *X* to obtain a watermarked image *X'*. *W* is of the same dimensions as the image *X*. If the watermarked image *X'* is then modified in any manner, the watermark *W'* extracted from it will not be the same as the original watermark *W*. Watermark insertion proceeds by examining each pixel *X(i,j)* in turn, and applying the watermark extraction function *D*. If the extracted watermark value is equal to the desired watermark value, *W(i,j)*, processing continues with the next pixel; otherwise, the current pixel value is adjusted until the extracted watermark value equals the desired value. This process is repeated for each pixel in the image, at the end of which the watermark has been completely inserted into the image.

The watermark extraction function is defined as:

$$W(i,j) = f(X(i,j)) \tag{1}$$

for grayscale images, and

$$W(i,j) = f_R(R(i,j)) \oplus f_G(G(i,j)) \oplus f_B(B(i,j)) \tag{2}$$

for RGB color images. Note that a similar extraction function can be defined for a different color space or for that matter a multi-band image. The functions $f(\cdot), f_R(\cdot), f_G(\cdot)$, and $f_B(\cdot)$, are implemented as binary lookup tables (hence we refer to them as the LUT's) , and $\oplus$ indicates an XOR operation. The LUT's are constructed from the secret key *K* known only to the transmitter and receiver (verifier). One way to do this, for example, is by using the key *K* to seed a pseudo-random number sequence that is used to define the mapping function implemented by the LUT's.

In addition to the basic technique described above, the Yeung-Mintzer technique also utilizes a modified error diffusion method to maintain proper average color over the image. 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 and checking it for any modifications. Hence, any changes made while arriving at this "original" image are not of interest to a potential attacker.

In earlier versions of the technique [14] Yeung and Mintzer use a binary logo as the watermark image *W*. This assists in visualizing changes made to the image after watermark extraction. However, in later versions of their

technique [15], they scramble the binary logo image prior to embedding in order to remove any structure in the watermark.. Again, since the attacks described in this paper assume no knowledge about the watermark, nor do they rely on any structure present within the watermark, this scrambling process is not relevant to our discussion and hence for the sake of brevity we choose to ignore it. We assume that the watermark is a binary logo and often refer to it in that manner.

Although the Yeung-Mintzer technique watermarks and subsequently authenticates image data in the spatial domain, the approach can also be used for image data in the transform domain. For example, if the image has been compressed using a DCT based technique like JPEG, the watermark can then be embedded in the DCT domain. Wu and Liu [17] propose one such technique that inserts a watermark only in the AC coefficients of a JPEG image. They take several additional measures to ensure that watermark insertion does not lead to visual degradation. For example, small coefficients are not modified to avoid high frequency distortions. Nevertheless, the essential structure of the method remains similar to the Yeung-Mintzer technique.

Clearly, there are some obvious advantages to the Yeung-Mintzer approach. First, if the watermark is a logo then it can carry some useful visual information about the 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 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 security. That is, how difficult is it for Oscar to lodge a substitution or impersonation attack? This question, as far as we know has not been properly addressed for most fragile watermarking techniques proposed in the literature, let alone the Yeung-Mintzer watermark. In the rest of this paper we present a detailed investigation of the security offered by the Yeung-Mintzer authentication watermark. We show that in its basic form the technique is vulnerable to different types of attacks. In Section 3 we show how if the same logo and key are reused for as few as two images, one could easily reconstruct the logo as well as the binary lookup table. In Section 4 we present another counterfeiting attack (the collage attack) and optimize its performance. The collage attack was introduced first by Holliman and Memon in [14] 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 by Holliman and Memon) can be significantly improved with the proposed

dithering process. Finally, in Section 5 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.

3. INFERRING THE SECRET BINARY FUNCTION AND THE WATERMARK LOGO

Given an image watermarked by the Yeung-Mintzer technique, how can Oscar successfully implement a substitution or impersonation attack? Clearly if Oscar knows the look-up tables f_R, f_G and f_B , then he can embed any watermark logo in any image. However, the look-up tables are constructed from the secret key that is unknown to Oscar. If the key space is not large then Oscar can try a brute-force attack, generating corresponding look-up tables corresponding to each key and checking if the extracted watermark is valid. For example, if the watermark is a logo then the key that results in look-up tables that extract a logo, has to be the secret key with very high probability as it is unlikely that a random look-up table can result in the extraction of a highly structured logo. However, if the logo is scrambled prior to insertion (say, using the same secret key), then there is no way for Oscar to know he has hit upon the right key by looking at the extracted watermark. Of course, all this can be prevented in the first place by choosing a sufficiently large (> 128) bit secret key.

Another way for Oscar to launch a successful substitution or impersonation attack is to somehow infer the look-up tables f_R, f_G and f_B without searching the key space. In this section, we show that this indeed can be done. That is, the look-up tables can be mostly inferred without knowledge of the secret key K and the binary watermark logo W, provided Oscar has access to at least two images that contain the same logo and have been watermarked using the same key.

Case 1: Grayscale Images. If the same logo lookup tables are used for multiple grayscale images, both the logo and the binary functions can be almost completely recovered from as few as two images. Given two grayscale $M \times N$ images U and V watermarked with the same secret key K and a binary logo W, one can write

$$W(i,j) = f(U(i,j)) = f(V(i,j)) \text{ for every pixel } (i,j). \tag{3}$$

This constitutes $M \times N$ equations for 256 unknowns $f(0), f(1), \ldots, f(255)$. Most of the equations are redundant and, depending on the image, there may not be a unique solution f. 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, $f(U(1,1)) = f(V(1,1))$, tells us that the values of f 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 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 $f() = 0$, the other to $f() = 1$, and several other sets for which the value of f 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. The system of equations (3) was then solved to recover the binary function f. 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. Combining the watermarked images 'Airfield' and 'Bridge' gave us 232 values for f. This is an unacceptable leakage of information, which clearly indicates that the same binary lookup table and 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 quickly saturates at 256.

Figure 2 Test image 'Lena'

Figure 3 Test image 'Airfield'

Figure 4 Test image 'Bridge'

Case 2: Color Images. A similar attack as the 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 $W(i,j)$, 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.

Actually it turns out that for the color case, instead of using an attack as described above, a more effective attack can be launched by directly solving a system of $M \times N$ linear equations in mod 2 arithmetic. In the rest of this section we detail how this can be done and then give experimental results showing that in a majority of cases the functions f_R, f_G, and f_B can be completely determined given just two color images containing the same (unknown) watermark logo.

Suppose an attacker has two different RGB color images X and Y with the pixel at location (i,j) denoted as $(X_R(i,j), X_G(i,j), X_B(i,j))$ and $(Y_R(i,j), Y_G(i,j), Y_B(i,j))$, respectively. Further, suppose that both X and Y contain the same watermark logo inserted using the same key K. Without loss of generality, suppose that X and Y are of the same size $M \times N$. Then we can express the logo $W(i,j)$ from both images as

$$f_R(X_R(i,j)) \oplus f_G(X_G(i,j)) \oplus f_B(X_B(i,j)) = f_R(Y_R(i,j)) \oplus f_G(Y_G(i,j)) \oplus f_B(Y_B(i,j)) \text{ for each pixel } (i,j). \quad (4)$$

To find the secret functions f_R, f_G, f_B and/or the binary logo one must solve the system of $M \times N$ equations (4) with 3×256 unknowns $f_R(0), f_G(0), f_B(0), f_R(1), f_G(1), f_B(1), \ldots, f_R(255), f_G(255), f_B(255)$.

By replacing the function f_R with its complement

$$f'_R(x) = 1 \oplus f_R(x),$$

the system (4) also holds because we can add ($\oplus$) binary 1 to every equation in (4). That is:

$$\{1 \oplus f_R(X_R(i,j))\} \oplus f_G(X_G(i,j)) \oplus f_B(X_B(i,j)) = \{1 \oplus f_R(Y_R(i,j))\} \oplus f_G(Y_G(i,j)) \oplus f_B(Y_B(i,j))$$
$$f'_R(X_R(i,j)) \oplus f_G(X_G(i,j)) \oplus f_B(X_B(i,j)) = f'_R(Y_R(i,j)) \oplus f_G(Y_G(i,j)) \oplus f_B(Y_B(i,j)).$$

By the same argument, we can replace f_G with $f'_G(x) = 1 \oplus f_G(x)$ and f_B with $f'_B(x) = 1 \oplus f_B(x)$. At the same time, the watermark W has to be replaced with its complement $1 \oplus W$ whenever we switch between f and f'. Consequently, the system (4) determines the functions f_R, f_G, f_B and the watermark W except for the above mentioned ambiguity. We note that it is possible to correctly authenticate an arbitrary image using either f or f' with the correct logo.

The arguments above also imply that one can set (arbitrarily) $f_R(255)=0$, $f_G(255)=0$, and $f_B(255)=0$. We can then rewrite the system (4) to get a new system of 3×255 unknowns in modulo 2 arithmetic:

$$f_R(X_R(i,j)) \oplus f_G(X_G(i,j)) \oplus f_B(X_B(i,j)) \oplus f_R(Y_R(i,j)) \oplus f_G(Y_G(i,j)) \oplus f_B(Y_B(i,j)) = 0 \text{ for each pixel } (i,j), \quad (5)$$

or in matrix notation

$$A\boldsymbol{f} = 0,$$

where $\boldsymbol{f} = (f_R(1), f_G(1), f_B(1), \ldots, f_R(255), f_G(255), f_B(255))$. The matrix A of this linear system is a binary matrix with 765 columns. The columns numbered $(3s-2)$, $s=1, 2, \ldots, 255$, correspond to the red channel, columns $(3s-1)$ correspond to the green channel, and columns $3s$, correspond to the blue channel. To solve the system of equations (5), we can use simple and straight-forward Gaussian elimination. This may turn out to be computationally prohibitive, hence we develop several mechanisms to make the algorithm run faster, as outlined below.

First observe that whenever two rows of A have their leftmost nonzero elements (which are equal to 1) in different columns they must be linearly independent. On the other hand, if two rows of A are linearly independent then they have their leftmost nonzero elements in different columns (if they do not, we can replace one of them with their sum modulo 2). This allows us to extract linearly independent rows of A in the following way:

1. Initialize matrix A to be a 765×765 matrix of all zeroes.
2. Selecting the pixels by rows, take the next pixel from the two authenticated images X and Y and write down the corresponding equation (5) that corresponds to this location *(i,j)*.
3. Find the k^{th} column where the leftmost 1 is located. If $A_{kk} = 0$, replace the k^{th} row in A with this row. If $A_{kk} \neq 0$, add the two rows together and repeat Step 3 with the summation (assuming the sum of rows is not equal to zero). If the sum is equal to a zero row, continue with Step 2 until all the pixels are used or the entire diagonal of A contains 1's.

The resulting matrix A will be a triangular matrix with zeros everywhere under the main diagonal. As the next step, a similar Gaussian-like elimination is applied to the second leftmost 1 in every row in A from the bottom to the top of the matrix A:

1. Starting with the bottom row, choose the next row.

2. Find the second leftmost 1 in the row. If it belongs to the l^{th} column add the l^{th} row to the current row. Repeat this step until either the l^{th} row has all zeros or there is no second leftmost 1. Continue with 1 till you go through all rows in A.

Finally, we repeat the same elimination for the third leftmost 1 in each row of A. As a result, we obtain a matrix similar to the one depicted in Figure 5 (but of far larger dimensions). If a row in A contains just one nonzero element (it has to be on the diagonal) then the corresponding variable must be equal to zero. From those rows, which contain more than two 1's, we cannot conclude anything and therefore we replace these rows with zeros. A row that contains exactly two 1's in two columns binds the two corresponding variables together saying that they are equal (both equal to one or zero). Consequently, a column that contains some 1's and a zero at the element that belongs to the main diagonal of A determines a group of variables of the same parity. Thus, we obtain a series of disjoint groups of colors with the same values of their binary functions in each group

	R1	G1	B1	R2	G2	B2	R3	G3	B3	R4	G4	B4	R5	G5	B5	R6	G6	B6	R7	G7	B7	R8	G8	B8	R9	G9	B9
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 5: A typical result for *A*.

.

If the number of pixels is significantly larger than the number of unknowns (which is the case even for two small 250×350 images), two very large groups and several (or none) significantly smaller groups are typically obtained. If the binary functions are essentially random without any bias to ones or zeros, it is highly unlikely that these two

disjoint groups would have the same value of their binary functions. So, we assign opposite values to the colors corresponding to both groups. The binary values of the smaller groups will remain undetermined. This is because the rank of the matrix A may be smaller than 3×255, or because some colors simply do not occur in either image. However, even if the binary functions are not known completely, the logo is often unique and can be recovered completely. It is also possible to create authenticated forgeries with an incomplete knowledge of the binary functions at the expense of sometimes having to make a large perturbation.

Number of unmatched pixels in the logo	Number of image pairs (out of 66 pairs)
0	64
6	1
4	1

Table 1: Results for logo reconstruction on database of 66 color (RGB) images with 250 x 350 pixels.

Number of undecided binary function values:	0	1	2	3	4	5	6	7	8	9	10
Occurrence (out of 66 pairs)	53	5	1	2	1	1	1	0	1	0	1

Table 2: Results for insertion function reconstruction on database of 66 color (RGB) images with 250 x 350 pixels.

Number of unmatched pixels in the logo	Number of image pairs (out of 66 pairs)
0	22
2	1
7	1
42	1
620	1

Table 3: Results for logo reconstruction on database of 66 color RGB images with 125 x 175 pixels.

In our tests, the original Matlab implementation ran about one hour (on a 333MHz Pentium II machine) for one pair of 250×350 images. Presorting the colors (in the three channels separately) by their frequency of appearance in both images speeds up the process significantly (1 to 6 minutes) because the two large groups of colors mentioned in the previous paragraphs are obtained faster. With increasing image dimensions, the running time increases only a little because all the equations with leftmost 1 in the k^{th} column are eliminated immediately once the main diagonal in A contains no zeroes from its k^{th} element to the end. We tested what portion of the logo and the secret functions f_R, f_G, f_B can be obtained from two small images with 250×350 pixels. We also ran tests for even smaller images of size 125×175 pixels. The results obtained from 66 pairs of color images are summarized in Tables 1, 2 and 3.

One of the surprising conclusions of this experiment is the observation that it is possible to recover a larger portion of the logo and the binary functions from color images than from grayscale images assuming the images have the same dimensions. This is because the system (5) for color images provides more "intertwined" constraints on the binary values than in the grayscale case. The complexity of the attack for the color case is not significantly larger than in the grayscale case.

4. THE COLLAGE ATTACK

The second problem described in this paper is actually common to watermarking schemes in which the watermark is embedded independently into local regions of 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 [16]. However, in their attack of the Yeung-Mintzer technique, Holliman and Memon had assumed the watermark logo is available to the attacker. We show here 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 quality 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 $X^1, X^2, \ldots, X^n$ authenticated with the same logo and binary functions, we can construct a new image (a collage) Y by selecting pixels from all n images, while preserving the relative location of the pixels in the images:

$$Y(i,j) = X^k(i,j) \text{ for some } k \in \{1, \ldots, n\} \text{ and all } (i,j). \tag{6}$$

The new image Y will be authentic if its authenticity is checked with the same logo and binary functions used for authenticating images $X^1, \ldots, X^n$. Given a large number of images $X^1, X^2, \ldots, X^n$ with RGB colors $(X^k_R(i,j), X^k_G(i,j), X^k_B(i,j))$, $k = 1, \ldots, n$, a pirate can take an arbitrary image Y with RGB colors $(Y_R(i,j), Y_G(i,j), Y_B(i,j))$, and create an approximation to Y by selecting the closest colors from all n images for every pixel

$$(Y_R, Y_G, Y_B) = (X^k_R, X^k_G, X^k_B) \text{ with } k = \min \arg_k \{ (Y_R - X^k_R)^2 + (Y_G - X^k_G)^2 + (Y_B - X^k_B)^2 \}, \tag{7}$$

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 $\| X - Y \|$ (in square norm). We show that this depends 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 Y. 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 randomly 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

$$E(D) = \int_0^a x.F'(x)dx = \int_0^a F_{\geq}(x)\,dx\,. \tag{8}$$

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

$$E(D,n) = \int_0^a F_{\geq}^n(x)dx\,. \tag{9}$$

The probability $F_{\geq}(x) = P(D \geq x) = \int_{z \in C} V(B(z,x) \cap C)\,dz = 1 - 4/3\ \pi x^3 + O(x^4)$, 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 for $E(D, n)$ for large n:

$$E(D,n) = \int_0^a F_{\geq}^n(x)dx \approx \int_0^a (1 - \frac{4}{3}\pi x^3 + O(x^4))^n dx\,. \tag{10}$$

The asymptotic behavior of this integral for large n can be obtained using the well-known Laplace technique [18, pp. 36]

$$E(D,n) \approx \frac{1}{3}\Gamma(\frac{1}{3})(\frac{3}{4\pi n})^{1/3} = \frac{1}{6}\Gamma(\frac{1}{3})\sqrt[3]{\frac{3}{\pi}}\sqrt[3]{\frac{1}{n}} = 0.554\sqrt[3]{\frac{1}{n}}\,. \tag{11}$$

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

$$\| X - Y \| = \frac{1}{MN}\sum_{i,j}[X(i,j) - Y(i,j)]^2 \approx 0.307\sqrt[3]{\frac{1}{n^2}}\,. \tag{12}$$

Improving Counterfeit Quality by Error Diffusion: It is possible to improve the visual appearance of the forgery by 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 color c_i' at pixel i can be expressed using the original color c_i as

$$\begin{aligned} c'_i &= c_i - decay \times error_i \\ error_i &= |\ closest\_color_{i-1} - c'_{i-1}\ |\,. \end{aligned} \tag{13}$$

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 color c. This happens when the color 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 randomly chosen color from a vertex. We have

$$P(D_v \geq x) = (1–4/3\,\pi x^3/8)\ ,\ \text{for}\ x \leq 1. \tag{14}$$

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

$$max(P(D_v \geq x)) = m < 1 \text{ for } 1< x \leq \sqrt{3} \tag{4}$$

to obtain

$$E(D_v, n) = \int_0^1 [1 - \frac{4}{3}\pi(x/2)^3]^n dx + O(m^n)\ . \tag{15}$$

Therefore

$$E(D_v, n) \approx \int_0^1 [1 - \frac{4}{3}\pi(x/2)^3]^n dx = 2\int_0^{1/2} [1 - \frac{4}{3}\pi t^3]^n dt = 2E(D, n)\ . \tag{16}$$

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

$$E(D_e, n) \approx \int_0^1 [1 - \frac{1}{4}\frac{4}{3}\pi x^3]^n dx = \sqrt[3]{4}\int_0^{1/\sqrt[3]{4}} [1 - \frac{4}{3}\pi t^3]^n dt = \sqrt[3]{4}E(D, n)\,, \tag{17}$$

$$E(D_f, n) \approx \int_0^1 [1 - \frac{1}{2}\frac{4}{3}\pi x^3]^n dx = \sqrt[3]{2}\int_0^{1/\sqrt[3]{2}} [1 - \frac{4}{3}\pi t^3]^n dt = \sqrt[3]{2}E(D, n)\ . \tag{18}$$

The accumulated error after processing K pixels in a row becomes

$$\begin{aligned} error_{i+K} \leq decay \times (\ error_{i+K-1} + decay \times (\ error_{i+K-2} + \ldots + decay \times\ error_i)) < \\ < decay \times error_{i+K-1} + decay^2 \times\ error_{i+K-2} + \ldots \end{aligned} \tag{19}$$

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

$$E(error_{i+K}) \leq \ decay/(1-decay) \times E(D_v, n) \text{ for all } K. \tag{20}$$

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 for *decay*:

$$\begin{aligned} decay/(1-decay) \times E(D_v, n) \leq \ & [\sqrt{3}-2\ E(D_v, n)]/2, \text{ or} \\ decay \leq 1-2/\sqrt{3}E(D_v, n). & \end{aligned} \tag{21}$$

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

$$decay = 1-2/\sqrt{3}E(D_v, n) = 1-\frac{4\beta}{\sqrt[2]{3}\sqrt[3]{n}}, \tag{22}$$

where

$$\beta = \frac{1}{6}\Gamma(\frac{1}{3})\sqrt[3]{\frac{6}{\pi}} \cong 0.554\,. \tag{23}$$

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.

5. CONCLUSION AND COUNTERMEASURES

There are several possibilities one may take to prevent the attacks described in Sections 3 and 4. The remedy suggested in [8] involved 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 look-up tables f_R, f_G and f_B to yield the watermark bit. In other words, we have

$$W\{i,j\} = f_R(XR(i,j)) \oplus f_G\{XG(i,j)) \ \oplus f_B(XB(i,j)) \ \oplus \ f_I(i) \oplus f_J(j). \tag{24}$$

Although this was effective against the attacks presented in [8], it does not prevent the attacks we presented in Section 4. We will now have a larger system of equations, which may require more than two images for gaining

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 6 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 in this Figure.

Forgery without diffusion (80 database images)

Forgery without diffusion (80 database images)

Forgery with diffusion (80 database images)

Forgery without diffusion (80 database images)

Figure 7 Examples of two images and forgeries obtained using the simple approach based on Eq. 4 (without error iffusion) 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 8 Examples of images and their forgeries obtained using a database of 300 images and error diffusion with a decay factor.

significant information about the watermark logo and the LUT. Also, the strategy is not effective against the collage attacks presented in section 4.

It appears that the simplest solution to deal with the collage attack would be to make the watermark depend on an image index. This would defeat 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 slightly 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 3 and 4. 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 3 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 security 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 only $f(p) = \text{Logo}(p)$ 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

J. Fridrich and M. Goljan were 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. N. Memon was partially supported by NSF grant NCR-9996145 and AFOSR Award Number F49620-01-1-0243.

REFERENCES

[1] Gus Simmons. "A Survey of Information Authentication," In *Contemporary Cryptography, The Science of Information Integrity*, IEEE Press, (1992).

[2] D. Stinson, *Cryptography, Theory and Practice*, CRC Press, (1995).

[3] R.L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126 (1978).

[4] R.L. Rivest, "The MD5 message digest algorithm." *Internet RFC 1321*, (1992).

[5] I. J. Cox and M. L. Miller. A review of watermarking and the importance of perceptual modeling. In *Proceedings, SPIE Human Vision and Electronic Imaging II*, SPIE Vol. 3016, (1997).

[6] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia Data Embedding and Watermarking Technologies," *IEEE Proceedings*, vol. 86, No. 6, pp 1064-1087, (1998).

[7] N. Memon and P. Wong. "Digital Watermarks. Protecting Multimedia Content," *Communications of the ACM*, 47(7):35-43 (1998).

[8] N. Memon, P. Wong and S. Shende. "On the Security of the Yeung-Mintzer Fragile Watermarking Technique." *Proceedings of PICS Conference*, Savannah, Georgia, (1999).

[9] P. W. Wong, "A watermark for image integrity and ownership verification," in *Proceedings of IS\&T PIC Conference*, (Portland, OR), 1998. Also available as Hewlett Packard Laboratories Technical Report HPL-97-72, (1997).

[10] P. Wong and N. Memon. Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification. *IEEE Transactions on Image Processing*, 10(10): 1593-1601 (2001).

[11] M. Schneider and S-F. Chang. "A robust content based signature for image authentication". in *Proceedings of ICIP*, (Lausanne, Switzerland), September (1996).

[12] S. Bhattacharjee, "Compression Tolerant Image Authentication", *Proceedings, Int. Conf. Image Proc.*,Chicago, Oct. (1998).

[13] J. Fridrich, "Image Watermarking for Tamper Detection", *Proceedings, Int. Conf. Image Proc.*, Chicago, Oct. (1998).

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

[15] M.~Yeung and F.~Mintzer, ``Invisible watermarking for image verification,'' *Journal of Electronic Imaging*, **7**(3), 576-591 (1998).

[16] M. Holliman and N. Memon. "Counterfeiting attacks for block-wise independent watermarking techniques." *IEEE Transactions on Image Processing*, 9(3):432-441 (2000).

[17] M. Wu and B. Liu, "Watermarking for Image Authentication", in Proceedin*gs of ICIP*, Chicago, IL, October (1998).

[18] A. Erdélyi, *Asymptotic Expansions*, , pp. 36, Dover Publications, Inc., New York, (1956).

FIGURE CAPTIONS

Figure 9: Authentication Model

Figure 10: Test image 'Lena'

Figure 11: Test image 'Airfield'

Figure 12: Test image 'Bridge'

Figure 13: A typical result for *A*.

Figure 14: 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 in this Figure.

Figure 15: Examples of two images and forgeries obtained using the simple approach based on Eq. 4 (without error iffusion) and with the diffusion and the optimal decay factor (with diffusion). Total of 80 database images were used in this Figure.

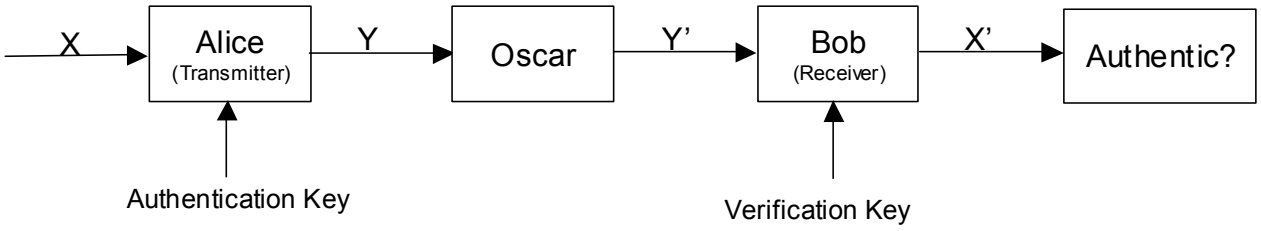
Figure 16: Examples of images and their forgeries obtained using a database of 300 images and error diffusion with a decay factor.

TABLE CAPTIONS

Table 4: Results for logo reconstruction on database of 66 color (RGB) images with 250 x 350 pixels.

Table 5: Results for insertion function reconstruction on database of 66 color (RGB) images with 250 x 350 pixels.

Table 6: Results for logo reconstruction on database of 66 color RGB images with 125 x 175 pixels.


X
Alice
(Transmitter)
Y
Oscar
Y'
Bob
(Receiver)
X'
Authentic?
Authentication Key
Verification Key

	R1	G1	B1	R2	G2	B2	R3	G3	B3	R4	G4	B4	R5	G5	B5	R6	G6	B6	R7	G7	B7	R8	G8	B8	R9	G9	B9
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	
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)**

 **Forgery without diffusion (80 database images)**	 **Forgery without diffusion (80 database images)**
 **Forgery with diffusion (80 database images)**	 **Forgery without diffusion (80 database images)**

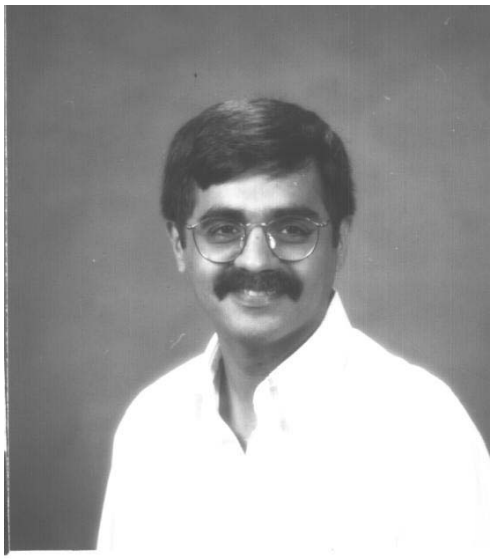
	
Original image	**Forgery with diffusion (300 database images)**
	
Original image	**Forgery with diffusion (300 database images)**
	
Original image	**Forgery with diffusion (300 database images)**

BIOGRAPHIES

Jessica Fridrich is a research professor at the Center for Intelligent Systems at State University of New York, Binghamton. In 1987, she received her MS degree in Applied Mathematics from the Czech Technical University in Prague, Czech Republic, and her PhD in systems science in 1995 from the State University of New York in Binghamton. Her main research interests are in the field of steganography and steganalysis, digital watermarking, authentication and tamper detection, and forensic analysis of digital images. In the last six years, Fridrich's research has been steadily supported by the US Air Force in the form of 13 research grants total worth over US$1.3mil, generating five US and international patents.

Miroslav Goljan is a post doctoral Research Associate in the Department of Electrical engineering at SUNY Binghamton. He received his MSc in Theoretical Informatics from Charles University in Prague in 1984 and his PhD in Electrical Engineering specialization on digital watermarking in December of 2001. His most recent contributions include the new paradigms of lossless watermarking for images, self-embedding, and development of dual-statistics steganalytic techniques.

Nasir Memon received his M. S. and Ph.D. from the University of Nebraska in 1989 and 1992 respectively. He is currently an Associate Professor in the Computer Science department at Polytechnic University, New York. He was a visiting Faculty at HP Labs Palo-Alto from August 1997 to July 1998 and From June to August 1999. Prof. Memon's research interests include Data Compression, Data Encryption, Image Processing, Multimedia Content Protection and Multimedia Communication and Computing. He has published more than 100 articles in journals and conference proceedings and holds two patents in image compression. He has been the principal investigator on funded research projects from HP, Intel, Panasonic, Mitsubishi, Sun Microsystems, AFOSR and NSF. In 1996 he received an NSF CAREER award for research in lossless image compression. He has organized and chaired many sessions in international conferences and is currently an associate editor for the IEEE Transactions on Image Processing and the ACM Multimedia Systems Journal.



From top to bottom: Jessica Fridrich, Miroslav Goljan,and Nasir Memon