On the Security of the Yeung-Mintzer Authentication Watermark

Nasir Memon[1]*, Sunil Shende*[2] *and Ping Wah Wong*[3]

[1]*Polytechnic University*
Brooklyn, NY 11201

[2]*Rutgers University*
Camden, NJ 01234

[3]*Hewlett-Packard Company*
Cupertino, CA 95014.

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. We show that under some conditions the Yeung-Mintzer technique is susceptible to impersonation attacks. We then propose some simple modifications to the technique that make it more robust against substitution and impersonation attacks.

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 has indeed originated from a specific source. This could be achieved with or without secrecy. In fact, for 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 Figure 1.

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 did originate from Alice and has not been modified, Alice computes an authenticated message Y which 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 information 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 $Y = (X \mid\mid a)$, where a is the appended authenticator and $\mid\mid$ denotes concatenation. Authentication could also be achieved by redundancy present in the structure of the message, which could be recognized by the receiver [7]. For ease of exposition, lets assume the former case.

If Bob receives $Y = (X \mid\mid a)$ he could verify, using a verification key, that a is indeed a valid authenticator for X and accepts the message. In a symmetric key system, the authentication and verification key are identical and both need to be kept secret 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 $Y' \neq Y$ with $Y' = (X' \mid\mid a')$ 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 [7]. One problem with the model described above is that Alice can always disclaim originating a message. Authentication techniques that are non-repudiable are also known. For an excellent recent survey on authentication techniques, the reader is referred to [7].

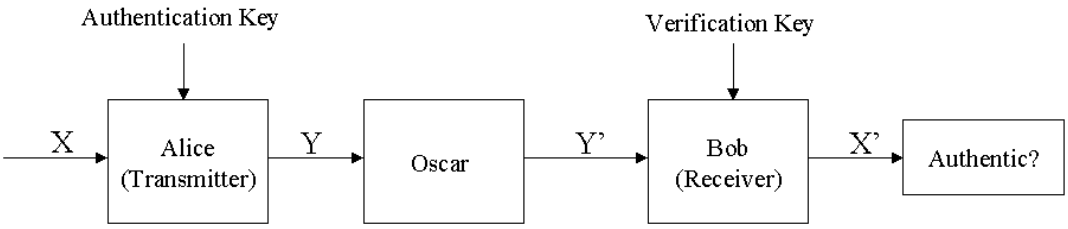

Figure 1: Authentication Model

Closely related to authentication techniques are digital signature schemes and message authentication code (MAC) generation algorithms. The former employs public key techniques to generate a signature for a message which can be verified by anyone having knowledge of the public key. Digital signature schemes are usually non-repudiable. MAC techniques are symmetric key (private key) based and in this sense similar to authentication codes. 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 [8].

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 has not been 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 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.

There have been many different watermarking techniques proposed in the literature and consequently there is a great deal of variation in how a watermark signal is embedded into an image. In general, the watermark insertion step can be represented as

$$X' = \mathcal{E}_K(X, W) \tag{1}$$

where X is the original image, W is the watermark information being embedded, K is the user's insertion key, and $\mathcal{E}$ represents the watermark insertion function. We adopt the notation throughout this paper that for an original image X, the watermarked variant is represented as X'.

Depending on the way the watermark is inserted, and depending on the nature of the watermarking algorithm, the detection or extraction method can take on very distinct approaches. One major difference between watermarking techniques is whether or not the watermark detection or extraction step requires the original image. Watermarking techniques that do not require the original image during the extraction process are called *oblivious* (or public) watermarking techniques. For oblivious watermarking techniques, watermark extraction works as follows:

$$\hat{W} = \mathcal{D}_{K'}(\hat{X}') \tag{2}$$

where $\hat{X}'$ is a possibly corrupted watermarked image, K' is the extraction key, $\mathcal{D}$ represents the watermark extraction/detection function, and $\hat{W}$ is the extracted watermark information.

In general, a watermark can be *visible* or *invisible*. Invisible watermarking schemes in turn can be classified as either *robust* or *fragile*. Robust watermarks are often used to prove ownership claims, and so are generally designed to withstand malicious attacks such as image scaling, cropping, lossy compression, and so forth. Examples of watermarking technique that are robust to such attacks are given in [1, 2, 3, 5]. In comparison, fragile watermarks are useful for purposes of authentication, and can potentially be used to verify the integrity of a given image's content. Fragile watermarks have been proposed in [10, 11, 12, 13, 14]. Overview and survey of watermarking techniques can be found in [4, 9, 6].

In Figure 2 we show the general framework in which a public key fragile watermarking technique [11] is used for authentication. As can be seen from the figure, authentication is achieved by embedding a binary logo into the image in such a manner that the visual quality of the image is unaffected. When the image needs to be authenticated or verified, the watermark (which is the binary logo) is extracted by using

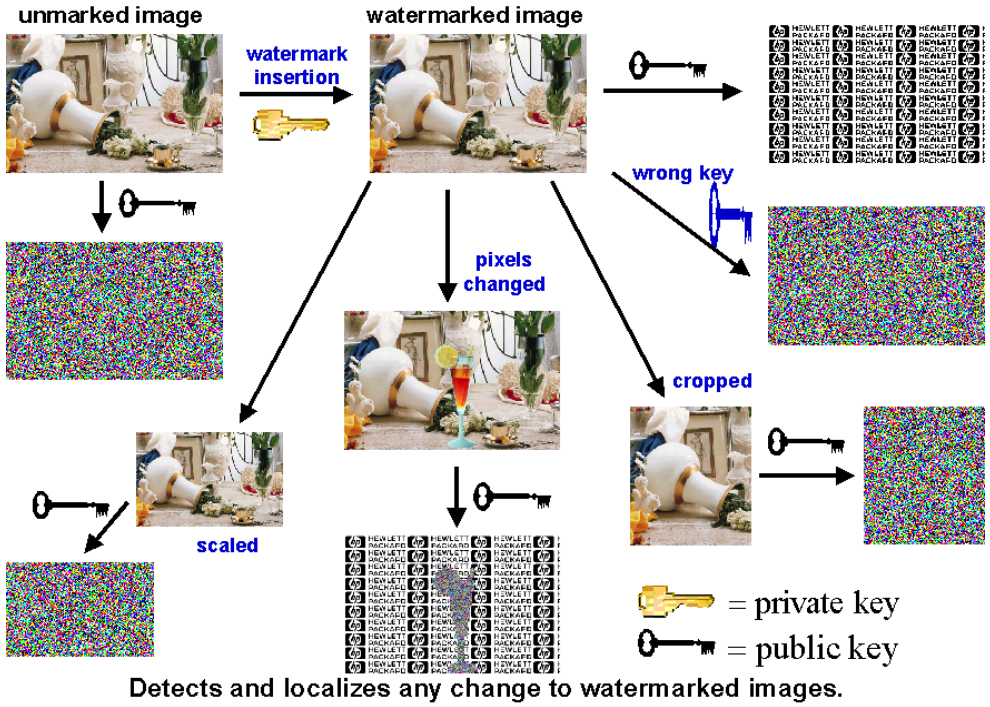

Figure 2: General framework of a fragile public key watermark [11] used for image authentication.

the unique key (public or private depending on the technique employed) associated with the source. The authenticity of the image is established through the integrity of the extracted logo image. Any change made to the image would result in corruption of the corresponding pixels in the extracted logo. Since a fragile watermark is essentially used for authentication purposes, we prefer to use the term authentication watermark in the remaining of this paper.

One advantage of using fragile watermarks for authentication, as opposed to a conventional authentication technique is that with fragile watermarking, the authenticator is inseparably bound to the content. This greatly simplifies the logistical problem of data handling and incorporating an authentication function in applications which represent images in one of the many possible different data formats that are used in practice today. Another advantage is that authentication watermarks allow the determination of the exact pixel locations where the image has been modified. Hence there has been considerable interest in developing fragile or authentication 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 fact, we show in this paper that some proposed techniques, have some potential weaknesses and under certain reasonable assumptions, are subject to substitution and impersonation attacks. The rest of the paper is organized as follows. In the next section we give a brief description of the Yeung-Mintzer watermarking technique and some simple extensions to it proposed later by Wu and Liu [12]. In section three we show how an adversary Oscar, with the knowledge of the binary logo image used for embedding can launch a substitution attack. In section four we then show how our attack can be made difficult by making some minor but crucial modifications to the originally proposed techniques.

The Yeung and Mintzer authentication watermark

Yeung and Mintzer [13, 14] recently proposed an authentication watermarking method to protect the integrity of images, in which a binary watermark image W is embedded into a source image X, so that subsequent alterations to the watermarked image X' should be detected. Generally W is a binary image of the same dimensions as the image X. However, W could have been created by tiling several copies of a smaller binary logo image similar to the case shown in Figure 2. Watermark insertion proceeds by examining each pixel $X_{i,j}$ in turn, and applying the watermark extraction function $\mathcal{D}$. 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.

The watermark extraction function is computed from the owner's key, and is defined as:

$$W_{i,j} = f_R(X_R(i,j)) \oplus f_G(X_G(i,j)) \oplus f_B(X_B(i,j)) \quad (3)$$

for RGB color images, and $W_{i,j} = f(X(i,j))$ for grayscale images, where the functions $f_R(), f_G()$ and $f_B()$ are binary lookup tables, one per color component, and $\oplus$ indicates an XOR operation. The lookup table contents are known only to a user possessing the key. For example, the key could be used to seed a pseudo-random number sequence used to generate the tables.

In addition to this process, since original pixel values are modified during watermark insertion, error diffusion is used 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 any modifications. Hence any changes made while arriving at this "original" image are not of interest to a potential attacker. In order to make substitution attacks difficult, Yeung and Mintzer also propose scrambling the binary watermark logo image using chaotic mixing techniques.

Image verification is accomplished by applying the watermark extraction function to X' to generate $\hat{W}$, which is compared to the original watermark W. Changes to any portion of the image X' should result in changes to the corresponding block of the extracted watermark. Clearly, watermark insertion and extraction are both extremely simple operations and can be implemented with low space and time complexity.

Although the above 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, then the watermark can be embedded in the compressed domain by restricting the insertion process to the DC coefficients only. That is, watermarking can be done by first extracting the DC coefficients and then using the corresponding insertion and verification process on the DC image. Restricting the watermark to DC coefficients may not be a good idea as an attacker may be able to make subtle modifications to an image while preserving the integrity of the embedded logo. Also, changing the DC coefficient in smooth regions can lead to the infamous blocking artifacts, commonly found in block DCT based compression techniques. Hence, Wu and Liu [12] insert the watermark only in the AC coefficients of the 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 the same as the Yeung-Mintzer technique.

Inferring Lookup Tables

Given an image watermarked by the Yeung-Mintzer technique, how can Oscar successfully implement a substitution attack? In the Yeung-Mintzer scheme, a secret key is used in a random number generator to populate the three tables $f_R(), f_G()$ and $f_B()$, each of size 256 bits. The key to our attack is to determine the content of the look up tables (a total of 768 bits).

If Oscar does not know the lookup tables f_R, f_G and f_B and the binary watermark logo W, a substitution attack appears to be very difficult. However, what if we assume that Oscar knows the watermark logo W? At first glance, it appears that the security of the scheme is still assured by the fact that Oscar must now correctly guess the correct lookup table combination from among $(2^{256})^3$ possibilities (each of the three functions in any combination can be independently fixed in 2^{256} ways). This is not true. We show in the rest of this section that there is, however, a way by which Oscar can attack the scheme successfully while examining far fewer possibilities. In particular, by analyzing a large enough sample of pixel values (RGB color triples) and their corresponding extracted watermark bits, Oscar can reduce the search space significantly that a brute force enumeration of candidate lookup functions becomes computationally feasible.

Consider an image X with associated watermark W. Let $X_i = (R_i, G_i, B_i)$, $i \geq 1$, denote the i^th color triple in X when the image is scanned in some order, and let W_i denote the corresponding extracted watermark bit. If f_R, f_G, f_B are the three unknown lookup table functions, we have:

$$W_i \quad = \quad f_R(R_i) \oplus f_G(G_i) \oplus f_B(B_i)$$

for each value of $i \geq 1$. Note that W_i is equal to 0 (or to 1) exactly when the unknown three-bit sequence $(f_R(R_i), f_G(G_i), f_B(B_i))$ has *even parity* (respectively, has *odd parity*). Thus, knowledge of W_i allows us reduce from 8 to 4 the number of choices for the unknown three-bit sequence. Next, consider a pair of color triples (R_i, G_i, B_i) and (R_j, G_j, B_j) in the source image. Naturally, if the triples are identical, then so are the corresponding watermark bits. Suppose that the two triples agree exactly on some pair of color values; say, $R_i = R_j$ and $G_i = G_j$. Then, choosing

arbitrary values $f_R(R_i)$ and $f_G(G_i)$ automatically fixes the values $f_B(B_i)$ and $f_B(B_j)$ because the right hand sides of the equations

$$f_B(B_i) = W_i \oplus f_R(R_i) \oplus f_G(G_i) \quad (4)$$
$$f_B(B_j) = W_j \oplus f_R(R_j) \oplus f_G(G_j) \quad (5)$$

are fully determined. The space of 16 possibilities for the lookup function values for the two triples is thereby reduced to only 4.

These observations motivate the construction of an undirected graph $G_2(X)$ associated with image X and its watermark W. For every distinct triple of color values in the image, there is an associated node in the graph $G_2(X)$. Edges in the graph are defined as follows: For any pair of triples that have identical color components in two corresponding positions, we create an undirected edge between their associated nodes (the subscript 2 in the notation $G_2(X)$ reflects this). Now, consider a *connected component* of $G_2(X)$, i.e. a subset of nodes with the property that from any node in the subset, we can reach every other node in the subset via a sequence of graph edges. From the discussion in the previous paragraph, specifically Equations (4) and (5), an inductive argument shows that *exactly four possible assignments* of lookup functions are consistent with the watermark bits associated with nodes in the connected component. Specifically, we can start at some arbitrary initial node in the component and choose two lookup function values for the triple corresponding to the node. The values for the remaining triples in the connected component can be completely inferred by following paths beginning at the initial node and using the above equations for each successive edge along any such path.

If the graph $G_2(X)$ has k connected components, the preceding paragraph would imply that the search space for inferring the actual lookup functions would have 4^k possibilities. In fact, the search space can be circumscribed even further: we create another graph $G_1(X)$ with the same nodes as in $G_2(X)$ but whose edges are between nodes whose corresponding color values share at least one identical color value. Thus, an edge in $G_2(X)$ is also in $G_1(X)$ but not vice-versa. If two nodes in different connected components of $G_2(X)$ are joined by an edge in $G_1(X)$, then the two components of $G_2(X)$ can be fully inferred by looking at only 8 possibilities and not $4^2 = 16$. In general, if k different connected components of $G_2(X)$ are joined together in one connected component of $G_1(X)$, then $4 \cdot 2^{k-1}$ lookup table choices cover all the possibilities.

Provide that Oscar has access to a large enough sample of the color triples and their associated watermark bits, the graphs $G_1(X)$ and $G_2(X)$ can both be expected to be richly connected (since many pixels would have at least one and often two color values in common). Our preliminary experimental results show that typical images have very few connected components, i.e., most pixels share one or two color component values with some other pixels in typical images. Consequently, Oscar can break the Yeung-Mintzer scheme by brute-force enumeration of the modest number of possible color lookup table possibilities, a computationally feasible operation under the circumstances.

Specifically, if the watermark image is known to Oscar, along with the source image, then he gains significant amount of information about the functions f_R, f_G and f_B. This is because every RGB triple that he can find that has two elements in common with a previously encountered triple, gives him more information about the table thereby enabling him to narrow his search space. In the terminology of the graph based framework, Oscar starts with a graph containing 256^3 components (one for every RGB triple). Let a non-trivial component be a component that contains more than one node. The degree of uncertainty that Oscar needs to uncover to attack this watermarking scheme is no more than $2^{256\times 3}$ or 2^{768}. Every time Oscar finds two triples that have two color quantities in common, he then checks to see if this pair of quantities are already connected by an edge in any of the existing non-trivial components. If they are not already connected, then Oscar forms a new edge and he has effectively reduced the degree of uncertainty by a factor of 2. Hence the size of the search space is also reduced by a factor of 2. His final search space is no more than 2^{768-E} where E is the total number of edges added, each causing the total number of components to shrink by one. Note also that any component in $G_2()$ can have a maximum size of 256, and hence the numebr of edges can be at most 255. As a result, we have in the best possible case a final search space of 4.

Thwarting a substitution attack

Our attack described in the previous section is based on the assumption that the watermark logo W is known. Before we consider further, we would like to note that neither Yeung and Mintzer [13, 14] nor Wu and Liu [12] explicitly mention in their paper that the logo image needs to be maintained as a secret. Nor do they address the issue of the security of their scheme given the fact that the watermark logo is known.

One may then ask whether assuming Oscar knows the watermark logo W is a reasonable assumption? In our opinion, it is. Consider an image merchant who puts an authentication watermark before transmitting the image to the buyers, it is likely that the merchant would use an logo representing the image retail company. Similarly, a news reporter who watermarks images is likely to use a logo that identifies the

reporter, or identifies the news organization. Similar arguments can be made for cases where the images are used for medical or legal purposes.

Nevertheless, irrespective of whether one considers the above assumption to be fair or otherwise, it would be clearly desirable if one can remove the requirement that the watermark logo W remain secret. Here we propose a simple modification to the Yeung-Mintzer authentication watermark (which can also be suitably adapted for the Wu-Liu watermark).

One way of preventing Oscar from achieving this reduction in the search space is by 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 XORed' 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

$$W_{i,j} = f_R(X_R(i,j)) \oplus f_G(X_G(i,j)) \oplus f_B(X_B(i,j) \oplus f_I(i) \oplus f_J(j)) \quad (6)$$

Clearly, Oscar now cannot reduce the search space by combining information form pixels at different spatial locations. Instead he needs to construct a different graph for each location his search space would then be exponential in the number of components in the union of all these graphs. For a 1000×1000 image this simple modification would increase his search space by an additional factor of 10^6 in the exponent. The cost for achieving this additional safety is two additional table look-ups (which can be done in parallel) and the storage for two additional tables, both of which can be kept to small size N (say, 1000) by performing an appropriate modulo N reduction prior to performing the table look-up.

References

[1] F. M. Boland, J. J. K. Ó Ruanaidh and C. Dautzenberg, "Watermarking Digital Images for Copyright Protection," *Proceedings of IEE International Conference on Image Procesisng and Its Applications* (Edinburgh), pp. 321-326, July 1995.

[2] G. W. Braudaway, "Protecting Publicly-Available Images with an Invisible Watermark," *Proceedings of IEEE International Conference on Image Processing* (Santa Barbara, CA), Oct. 1997.

[3] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, 1997.

[4] I. J. Cox and M. L. Miller. "A review of watermarking and the importance of perceptual modeling." *Proceedings of SPIE Human Vision and Electronic Imaging II*, vol. 3016, February 1997.

[5] E. Koch and J. Zhao, "Towards Robust and Hidden Image COpyright Labeling," *Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image Processing* (Neos Marmaras, Halkidiki, Greece), June, 1995, pp. 452-455.

[6] N. Memon and P. W. Wong, "Protecting Digital Media Content," *Communications of the ACM*, vol. 41, no. 7, July 1998.

[7] G. Simmons. "A Survey of Information Authentication," *In Contemporary Cryptography, The Science of Information Integrity*, IEEE Press, 1992.

[8] D. Stinson, *Cryptography, Theory and Practice*, CRC Press, 1995.

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

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

[11] P. W. Wong, "A public key watermark for image verification and authentication," *Proceedings of IEEE International Conference on Image Processing* (Chicago, IL), October 1998.

[12] M. Wu and B. Liu, "Watermarking for image authentication," *Proceedings of IEEE International Conference on Image Processing* (Chicago, IL), October 1998.

[13] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 680–683, October 1997.

[14] M. Yeung and F. Mintzer, "Invisible watermarking for image verification," *Journal of Electronic Imaging*, 7(3), 576-591, July 1998.