Efficient Oracle Attacks on Yeung-Mintzer and Variant Authentication Schemes*

Jinhai Wu[1], Bin B. Zhu[2], Shipeng Li[2], Fuzong Lin[1]
[1]*State Key Lab of Intelligent Tech. & Systems, Tsinghua Univ., Beijing, 100084, P. R. China*
[2]*Microsoft Research Asia, Beijing, 100080, P. R. China*
wu-jh01@mails.tsinghua.edu.cn, {binzhu, spli}@microsoft.com, linfz@mail.tsinghua.edu.cn

Abstract

The Yeung-Mintzer (Y-M) image authentication scheme has been well studied. Several vulnerabilities and modified schemes to fix them have been reported. In this paper, we propose a novel oracle attack on the Y-M scheme and its variations. Our attack is very different from the previously proposed attacks. A single authenticated image plus access to a verifier (oracle) is enough in our attack. The verifier returns if a testing image is authentic or not. Locations of tampered pixels are not needed. To launch the attack, a single pixel is modified and the resulting image is sent to the verifier. Observation of outputs of the verifier is used to deduce the secret mapping functions and the embedded logo within an uncertainty of two possibilities. The deduced mapping functions are then used to modify the content of an authenticated image without detection or to authenticate an arbitrary image of the same size. Note that the logo is not used in the forgery so sophisticated protection of the logo cannot thwart the attack. Our attack is very efficient. Only 255 trials are needed to attack an 8-bit grayscale image and 765 trails for a 24-bit color image. The proposed attack can also be applied to attack pixel-wise variations of the Y-M scheme proposed to fix the previously reported vulnerabilities.

1. Introduction

Multimedia authentication is a technology to verify authenticity and integrity of multimedia signals. It has been actively studied in recent years. Proposed technologies are reviewed in [1][2]. One of them is the Yeung-Mintzer (Y-M) scheme which embeds a binary logo image to detect manipulations to individual pixels [3][4]. For grayscale images, the scheme uses a secret binary function f to enforce the following relationship between an image I, possibly modified if necessary, and a binary logo image L:

$$L(i,j) = f(I(i,j)), \forall\ all\ pixel\ (i,j) \quad (1)$$

For RGB color images, three mapping functions f_R, f_G, and f_B are applied to each color component respectively to enforce the following relationship:

$$L(i,j) = f_R(I_R(i,j)) \oplus f_G(I_G(i,j)) \oplus f_B(I_B(i,j)) \quad (2)$$

Authenticity is verified by applying the same mapping function(s) to a challenged image and comparing the result with the logo image. Any discrepancy indicates inauthenticity of the image and tampered pixels are located by disagreed bits.

The logo image has to be kept secret, otherwise the secret mapping functions can be deduced [5]. If the logo is unknown but the logo and the mapping functions are reused, two such authenticated images are needed on average to deduce about 90% of the mapping function for grayscale images [6][7]. It is much more difficult to successfully launch this attack on color images. The vector quantization attack proposed in [8] can also be used to authenticate an arbitrary image under the aforementioned scenarios.

In this paper, the security of Y-M scheme and several modified schemes are examined again from a different perspective. We assume that one authenticated image and an oracle, i.e., a verifier which is a program to check authenticity, are available. It turns out that there exists a very efficient oracle attack to these schemes. The attack needs 255 trials for 8-bit grayscale images and 765 trials for 24-bit color images of arbitrary sizes. Without loss of generality, grayscale and color images are assumed to be 8 and 24 bits in depth, respectively.

The rest of this paper is organized as follows. In Section 2, we describe our proposed oracle attack on the Y-M scheme in detail. Attacks on modified Y-M schemes and comparison with previously proposed attacks are described in Section 3. Experiments are reported in Section 4 before we conclude the paper.

*This work was done when Jinhai Wu was an intern at Microsoft Research Asia.

2. Efficient oracle attack on Y-M scheme

A secret binary function f in the Y-M scheme is applied to a color component of each image pixel to map an integer in [0, 255] to either 0 or 1. There are 2^{256} possible binary functions for each color component. Instead of deducing f from so many possibilities as in [5] or to solve many equations to find out f as in [6][7], we exploit the fact that the same mapping function is applied to each pixel and mapping functions for different color components are independently applied. For a mapping function, its corresponding color component of one pixel is modified at a time and oracle outputs are used to group input integers into two disjoint sets corresponding to the function's two respective output values. Integers in the same set can be exchanged without detection by the verifier. There is no need to know the exact output value associated with each set to modify the authenticated image yet pass the verifier or to authenticate an arbitrary image. The secret logo can also be deduced with uncertainty of two possibilities.

In this section, we first describe our oracle attack on the Y-M scheme for grayscale images. Then the attack is extended to RGB color images. The last subsection describes how to infer the secret logo.

2.1. Oracle attack on grayscale images

We first select an arbitrary pixel, for example the pixel $I(1,1)$ without loss of generality, from an authenticated image I. The two sets S_P and S_F are initialized as: $S_p = \{I(1,1)\}$ and $S_F = \{\}$. The following procedure is applied to find out the members of S_P and S_F:

For $i = 0$ to 255, $i \neq Original\ I(1,1)$ {

1. Set $I(1,1) = i$ and send the resulting image to the verifier and observe the output T_i
2. If $T_i =$ Pass, then $S_P = S_P \bigcup \{i\}$; otherwise, $S_F = S_F \bigcup \{i\}$.

}

At the end of the procedure, all the integers in [0, 255] are partitioned into the two disjoint sets S_P and S_F. The total number of testing by the verifier is 255, a number so small that the aforementioned procedure is feasible even when the verifier is controlled by a trusted party. In such a case, an attacker or a small number of cooperated attackers submit testing images to an oracle and observe the outputs.

Once S_P and S_F are found, they can be used to modify the content of the image without detection by the verifier or counterfeit a valid authentic image for an arbitrary image of the same size. To modify the content of an authentic image, each modified pixel undergoes the following procedure: the pixel value of the authentic image and the corresponding set are first found. The pixel is then set to the value in the same set closest to the desired value. It is apparent that the verifier cannot detect such a modification. Error diffusion used in the Y-M scheme can be employed in this process to reduce perceptual distortion.

The obtained information can also be used to authenticate an arbitrary image of the same size: Each pixel of an arbitrary image is set to a value inside the set which contains the pixel value at the same position of the authentic image. The same procedure used in the watermarking process of the Y-M scheme can be used to choose such a value. The resulting image bears the same logo as the authentic image and will be certified as authentic by the verifier.

This attack is simple and useful to attackers since it can at least cast doubts on the validity of any testing results reported by the Y-M authentication scheme.

2.2. Oracle attack on RGB color images

For RGB color images, the same grouping procedure described in Section 2.1 is applied to each color component. Only one color component is modified for each trial. For example, we can modify $I_R(1,1)$ while keeping $I_G(1,1)$ and $I_B(1,1)$ unchanged to obtain the two sets S_{RP} and S_{RF}. At the end of the oracle attack, six sets are obtained: S_{RP}, S_{RF}, S_{GP}, S_{GF}, S_{BP} and S_{BF}, with two sets for each color component. The total number of trials is $255 \times 3 = 765$ which is small enough to make this attack feasible even when the verifier is controlled by a trusted party. In a similar way as in the grayscale image case, these sets can be used to modify the content of an authentic image without detection or to authenticate an arbitrary image of the same size.

2.3. Inferring secret logo

With the sets obtained above, the secret logo can be inferred with an uncertainty of two possibilities. We emphasize here that this inferred logo is *not* used in launching a successful forgery as described above. It is straightforward for grayscale images. For color images, each color component has two possibilities, and Eq. (2) gives $2^3 = 8$ possibilities for the secret binary logo L. These 8 possibilities can be further reduced to 2 with the following method. A pixel of the logo image L, for example $L(1,1)$ without loss of

generality, is selected and we calculate the results of $L(1,1)$ XORed with all the other pixels $L(i,j)$. From Eq. (2), we have:

$$L(i,j) \oplus L(1,1) = f_R(I_R(i,j)) \oplus f_R(I_R(1,1)) \oplus f_G(I_G(i,j)) \oplus f_G(I_G(1,1)) \oplus f_B(I_B(i,j)) \oplus f_B(I_B(1,1)).$$

The value $L_R \equiv f_R(I_R(i,j)) \oplus f_R(I_R(1,1))$ can be calculated with the sets S_{RP} and S_{RF}. If $I_R(i,j)$ and $I_R(1,1)$ are in the same set, then $L_R = 0$; otherwise $L_R = 1$. The XORing result for the other two color components can be similarly found. Therefore the result $L(i,j) \oplus L(1,1)$ can be obtained for each pixel (i,j). This implies that the secret logo value at any pixel can be uniquely deduced once the value of $L(1,1)$ is known. There are only two possible values for $L(1,1)$, we conclude that the secret logo can be deduced to an uncertainty of two possibilities.

3. Attacks on modified Y-M schemes

As we mentioned in Section 1, several vulnerabilities of the Y-M scheme have been reported under different scenarios. Several modified schemes have been proposed to thwart these attacks. These schemes can be classified into two categories: pixel-wise schemes and neighborhood-dependent schemes. The former maintains relative independency of pixels in mapping each pixel to a logo bit, while the latter introduces dependency on previously processed neighboring pixels in mapping the current pixel to the corresponding logo bit. Our proposed oracle attack can successfully attack schemes of the first category but is ineffective to attack the schemes of the second category.

3.1. Pixel-wise schemes

Yeung et al. [4] proposed a modified scheme which scrambles the logo before embedding to thwart attacks that need the knowledge of the logo such as the one proposed in [5]. Lu et al. [9] proposed a similar scheme to thwart the quantization attack by breaking the known position relationship between the watermarked image and the logo. Our attack works for these schemes since our attack does not need the knowledge of the logo. The logo can no longer be deduced but we can still modify image content yet pass the verifier or authenticate an arbitrary image.

Memon et al. [5] modified Eq. (2) to the following by introducing position dependency to thwart their own attack to deduce the mapping functions with known logo:

$$L(i,j) = f_R(I_R(i,j)) \oplus f_G(I_G(i,j)) \oplus f_B(I_B(i,j)) \oplus f_I(i) \oplus f_J(j), \quad (3)$$

where f_I and f_J are two additional binary mapping functions to map horizontal and vertical axis indices to either 0 or 1. It has been shown that such a modification increases the search space by an exponential factor to make it extremely difficult to infer the mapping functions [5]. Our proposed oracle attack can effectively attack this modified scheme except inferring the logo since Eq. (3) can be expressed as:

$$L'(i,j) \equiv L(i,j) \oplus f_I(i) \oplus f_J(j) = f_R(I_R(i,j)) \oplus f_G(I_G(i,j)) \oplus f_B(I_B(i,j)).$$

Note that the right side is exactly the same as Eq. (2). We do not need to deduce the mapping functions f_I and f_J to mount our oracle attack. Making the mapping functions and the logo depend on a unique image index as proposed in [6][7] does not have any effect on the proposed oracle attacks either.

Zhong et al. [10] proposed a scheme that partitions pixels into disjoint watermarking and feature subspaces which can be public information. The mapping functions are generated from hashed features extracted from the feature subspace and used to map pixels in the watermarking subspace. This scheme tries to avoid the same mapping functions being used by two authenticated images to thwart the attack proposed in [6][7]. When the subspace partition is known, the scheme has no impact to the effectiveness of our proposed attack. If the partition is not known, more trials are needed to deduce the partition. If the verifier indicates tampered pixels in addition to pass or failure, this can be easily done by modifying each pixel and observe tampered pixels reported by the verifier. A pixel in the watermarking subspace can cause only the pixel itself to be reported to be tampered while a pixel in the feature subspace can causes many pixels to be reported as tampered. This difference is used to find out the partition.

We have developed a pixel-wise scheme that can effectively thwart our oracle attacks as well as the previously proposed attacks yet maintain fine tamper localization. The scheme is to be reported separately.

3.2. Neighborhood-dependent schemes

Dependency of a mapping function on processed neighboring pixels can be introduced to thwart the previously reported attacks. For example, Fridrich et al. [11] replaces the mapping function with the parity of the ciphertext after a block cipher is applied to a block that includes the current pixel and processed

neighboring pixels. A similar scheme is proposed in [12]. In these schemes, different mapping functions are applied to different pixels, and therefore render our attack ineffective. However, the improved security is achieved at the cost of weakened tamper localization.

4. Experiments

We did an experiment to show our proposed oracle attack. Fig. 1 shows an image authenticated by the Y-M scheme. Fig. 2 shows the embedded logo. There are four Chinese characters at the center of the image. The two Chinese characters on the diagonal of 45 degree were then switched and resulted in a different meaning. The resulting image is shown in Fig. 3 which looks authentic, but the resulting logo, shown in Fig. 4, by applying the Y-M scheme indicates that the two characters are tampered, as expected. We then launched our oracle attack to modify Fig. 3 to pass the verifier. The resulting image is shown in Fig. 5, and the corresponding logo from the Y-M scheme is shown in Fig. 6, which is identical to Fig. 2 of the authentic image. In other words, the content-modified image shown in Fig. 5 passes the Y-M authentication scheme.

Fig.1: Authentic image

Fig. 3: Modified Image before oracle attack

Fig. 5: Faked image after oracle attack

Fig. 2: Embedded logo

Fig. 4: Extracted logo before the oracle attack

Fig. 6: Extracted logo after the oracle attack

5. Conclusion

In this paper, we have examined vulnerabilities of the Y-M authentication scheme and its variations with a more realistic assumption than other proposed attacks that only one authenticated image and a verifier are available. Our oracle attacks have then been described. These attacks are very efficient and effective for the Y-M scheme and its pixel-wise variations, but ineffective for the neighborhood dependent variations. Experiments on the proposed oracle attacks have also reported in this paper.

References

[1] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When Seeing Isn't Believing," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 40-49, March 2004.

[2] B. B. Zhu and M. D. Swanson, "Multimedia Authentication and Watermarking," *Multimedia Information Retrieval and Management*, D. Feng, W. C. Siu, and H. Zhang, Eds. Springer-Verlag, 2003, Ch. 7, pp. 148-177.

[3] M. M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," *IEEE Int. Conf. Image Processing*, 1997, vol. 2, pp. 680-683.

[4] M. M. Yeung and F. C. Mintzer. "Invisible Watermarking for Image Verification," *J. Electronic Imaging*, vol. 7, no. 3, pp. 578-591, July 1998.

[5] N. Memon, S. Shende, and P. Wong, "On the Security of the Yeung-Mintzer Authentication Watermark," *Proc. IS&T PICS Symp.*, Savannah, Georgia, pp. 301-306, March 1999.

[6] J. Fridrich, M. Goljan, and N. Memon, "Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme," *Proc. SPIE vol. 3971 Security and Watermarking of Multimedia Contents II*, San Jose, CA, pp.428–437, January 2000.

[7] J. Fridrich, M. Goljan, and N. Memon, "Cryptanalysis of the Yeung-Mintzer Fragile Watermarking Technique," *J. Electronic Imaging*, vol. 11, pp.262-274, 2002.

[8] M. Holliman and N. Memon, "Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes," *IEEE Trans. Image Processing*, vol.9, no.3, pp. 432-441, March 2000.

[9] H. Lu, R. Shen, and F. Chung, "Fragile Watermarking Scheme for Image Authentication," *Electronics Letters*, vol. 39, no.12, pp. 898-900, June 2003.

[10] H. Zhong, F. Liu, and L. C. Jiao, "A New Fragile Watermarking Technique for Image Authentication," *Int. Conf. Signal Processing*, vol.1, pp. 792-795, Aug. 2002, Beijing.

[11] J. Fridrich, M. Goljan, and A. C. Baldoza, "New Fragile Authentication Watermark for Images," *IEEE Int. Conf. Image Processing*, vol. 1, pp. 446-449, Vancouver, Canada, Sept., 2000.

[12] C. T. Li, F. M. Yang, and C. S. Lee, "Oblivious Fragile Watermarking Scheme For Image Authentication," *IEEE Int. Conf. Acoustics, Speech, & Signal Processing*, Orlando, FL, USA, vol. VI, pp. 3445-3448, May 2002.