SECURE SEMI-FRAGILE WATERMARKING FOR IMAGE AUTHENTICATION

Chuhong Fei [a], *Raymond Kwong* [b], *and Deepa Kundur* [c]

[a] A.U.G. Signals Ltd., 73 Richmond St. W, Toronto, ON M4H 4E8 Canada
[b] University of Toronto, ECE Department, 10 King's College Road, Toronto, ON M5S 3G4 Canada
[c] Texas A&M University, ECE Department, 3128 TAMU College Station, TX, 77843-3128 USA

ABSTRACT

This paper proposes a secure semi-fragile authentication watermarking algorithm for natural images by embedding two complementary watermarks for content change analysis. Two authenticator watermarks are generated and embedded in different regions of the images: one for detecting malicious modifications and the other for estimating the degree of the changes. The proposed scheme is able to distinguish common content-preserving changes from malicious content-changing modifications. Simulations on real images demonstrate the effectiveness of the authentication watermarking scheme.

Index Terms— Multimedia authentication, Digital watermarking, Semi-fragile authentication.

1. INTRODUCTION

Many research efforts have been made on designing practical semi-fragile authentication schemes for natural images. Natural images such as journalist photos, typically contain objects which define the visual content of the images. Semi-fragile authentication should be able to distinguish incidental changes from malicious tampering attacks. Incidental changes are content-preserving manipulations, which include image compression or transcoding, filtering, and other common noises. These modifications keep the content recognizable although sometimes the visual quality may be degraded. In contrast, malicious attacks are content-changing manipulations, which include removing image objects and adding new objects. Such malicious modification could result in an image with a totally different meaning. A malicious attacker may actively exploit vulnerabilities of the authentication algorithm to produce a modified image which does not preserve the original visual meaning but can be wrongly authenticated.

Most of the existing work measures the similarity or correlation of the extracted watermark and the given watermark, then compares it with a given threshold to decide if the distortion is incidental or malicious. However, such measurement on the degree of overall distortion is not sufficient to detect active malicious tampering which could occur in a very small portion of the image. This is because the cumulative impact of an incidental distortion in the entire image could be more severe than a local malicious tampering. Some existing semi-fragile work [1, 2] assumes that malicious tampering occurs in only a local area, thus employs a semi-fragile scheme to detect the image area which is modified. This approach fails to address active malicious tampering in which the attacker can first apply an incidental distortion on the entire image and then modify certain image objects. In this way, the affected area is global but the malicious attack is concealed behind an incidental change. Another approach is just to detect the change in the highest order bits of the image pixels [3] since incidental distortions are unlikely to change the most significant bits. However, the approach leads to a security vulnerability since the attacker can modify less significant bits without being detected, and hence modify the image content also.

To distinguish incidental and malicious attacks, we employ two authenticators; one is a cryptographic authenticator to detect content changes, and the other is a smooth authenticator to estimate overall degree of changes. The cryptographic authenticator is employed to detect the amplitude of local changes, which is the key step to separate malicious and incidental changes. We use the other smooth authenticator to measure the impact of the changes in local areas for tampering localization. In order to locate the area of possible distortion, the second authenticator is independently generated in every local block. Using these two authenticators, we are able to not only detect malicious tampering on image content, but also to measure the degree of distortion locally. In addition, our proposed scheme is secure against malicious attacks on even a small area of an image, and is able to measure the degree of distortion and determine the location.

2. OUR PROPOSED ALGORITHM

2.1. Our Approach and System Diagram

Based on the above design ideas, we propose our approach as shwon in Fig. 1. We employ the 8×8 discrete cosine transform (DCT) and transform an image to the DCT domain similar to JPEG compression. The quantized values of the DC coefficients are extracted to represent salient image features. Then a message authentication code (MAC) is generated from the image features and embedded in middle DCT frequency

bands. In order to be forgiving of incidental changes, the values of DC coefficients are possibly modified to create a dead zone so the same features are extracted even under incidental changes. The second authenticator generation is performed independently on all 8×8 blocks. In each 8×8 image block, the DCT coefficients are quantized to compute a probabilistic checksum, called approximate message authenticator code (AMAC) [3]. The generated AMAC is then embedded in the middle or low frequency coefficients in the same block. Both AMAC generation and embedding are carried out independently on all DCT blocks, so possible incidental or malicious distortion in each block can be localized and the degree of distortion is measured locally by comparing the embedded AMAC watermark and the generated AMAC.

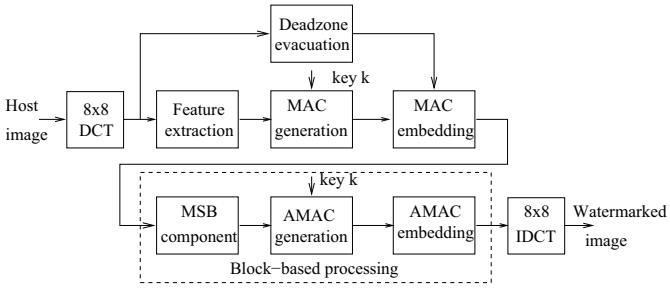


Fig. 1: Proposed authentication scheme.

2.2. MAC Generation and Embedding

2.2.1. Feature Extraction and MAC generation

To assure robustness under acceptable manipulations and fragility under malicious manipulations, the DC component of the DCT coefficients is sought to represent the visual meaning of an image. For an image I of size $M \times N$, let $C_{u,v}(i,j)$ denote the DCT coefficients at frequency band (i,j) for $1 \leq i,j \leq 8$ in DCT block (u,v) $1 \leq u \leq \lceil M/8 \rceil, 1 \leq v \leq \lceil N/8 \rceil$. The content feature is extracted from the DC component $C_{u,v}(1,1)$ as follows,

$$F_{u,v} = \left\lfloor \frac{C_{u,v}(1,1)}{\Delta_c} \right\rfloor \text{ for } 1 \leq u \leq \left\lceil \frac{M}{8} \right\rceil, 1 \leq v \leq \left\lceil \frac{N}{8} \right\rceil \tag{1}$$

where $\lfloor x \rfloor$ denotes the floor function, which gives the largest integer less than or equal to x, and Δ_c is the quantization step for content extraction. In our implementation, the step size Δ_c is set to be 256 for the DC coefficients in the range of $[0, 2048)$. Our experiments show that such choice of Δ_c makes it almost infeasible for the attacker to construct a meaningful fraudulent image without changing the extracted content feature. The attacker may damage the quality of the authenticated image by slightly changing the DC values in the same quantization interval but such quality change will also be detected in the second watermark in Section 2.3.

A traditional message authentication code (MAC) is applied on the feature sequence in binary representation $F_{u,v}, 1 \leq u \leq \lceil M/8 \rceil, 1 \leq v \leq \lceil N/8 \rceil$ using secure hashing algorithms such as HMAC based SHA-1, or SHA-256 algorithm with the use of the key k.

2.2.2. Dead-zone Evacuation

From the feature extraction and MAC generation process, it is possible to have a totally different authenticator even when the DC component is slightly changed. This happens when $C_{u,v}(1,1)$ is around quantization segment boundary $i\Delta_c$ for some integer i. To improve the robustness of the content feature, we introduce a dead-zone evacuation strategy by creating a gap around the segment boundary. Let T_d be the maximum allowable change for the dead zone and $T_d \leq \Delta_c/2$. We prohibit the DC coefficient values in the dead zone around the segment boundary. Given a DC coefficient $C_{u,v}(1,1)$, we change $C_{u,v}(1,1)$ to the nearest point outside the evacuation zone as follows,

$$\hat{C}_{u,v}(1,1) = \begin{cases} \left\lceil \frac{C_{u,v}(1,1)}{\Delta_c} \right\rceil * \Delta_c + T_d & \text{if } 0 \leq e \leq T_d \\ \left\lceil \frac{C_{u,v}(1,1)}{\Delta_c} \right\rceil * \Delta_c - T_d & \text{if } -T_d \leq e < 0 \\ C_{u,v}(1,1) & \text{Otherwise} \end{cases} \tag{2}$$

where $e = C_{u,v}(1,1) - [C_{u,v}(1,1)/\Delta_c] * \Delta_c$ is the quantization error. In our implementation, $T_d = 20$, which provides both a slight evacuation distortion with $PSNR = 53.22\,\text{dB}$ and a sufficiently wide dead zone with gap width 40.

2.2.3. MAC Embedding

Denote the generated MAC bits by $(m_1, m_2, \ldots, m_{L_m})$ where L_m is the length. Now we embed these MAC bits to DCT coefficients. In order to increase robustness of the embedded watermark, we adopt a Spread-Transform-Dither-Modulation (STDM) embedding scheme [4], which is a binary QIM scheme in the coefficients projected to a given spread spectrum vector. This scheme retains the security advantage of spread spectrum method and robustness advantage of QIM scheme over other one-bit embedding schemes.

To reduce the embedding distortion, we embed the MAC bits in the middle frequency bands. In each 8×8 block, we choose the frequency band $(4,4)$ for MAC embedding in our implementation. Therefore, for an image of size $M \times N$, there are $\lceil M/8 \rceil \lceil N/8 \rceil$ coefficients available for embedding. Since there are L_m bits of MAC authenticator, we can embed the MAC authenticator with one bit in N_m coefficients where $N_m = \lfloor \lceil M/8 \rceil \lceil N/8 \rceil / L_m \rfloor$. Table 1 describes the detailed algorithm of embedding one bit into N_m coefficients using STDM scheme. In our experiment for images of size 512×512, $L_m = 400$, $N_m = 10$ and $\Delta_m = 40$, the distortion due to MAC embedding is evaluated by $PSNR = 45.84$, which is acceptable.

Embedding Algorithm: Embed one bit m in L_m coefficients, $E_1, E_2, \ldots, E_{L_m}$ with key k.

1. First, a pseudorandom antipodal binary sequence $w_1, w_2, \ldots, w_{N_m}$ is generated by the key k where $w_i = \pm 1, i = 1, 2, \ldots, N_m$. The sequence is used as a spread spectrum sequence to correlate with the coefficients
$$R = \frac{1}{N_m} \sum_{i=1}^{N_m} w_i E_i. \quad (3)$$
2. The watermark bit m is embedded in R using the standard QIM scheme. The watermarked signal is given by
$$R^m = \left[\frac{R - d(m)}{\Delta_m} \right] \Delta_m + d(m) \quad (4)$$
where Δ_m is the quantization step for MAC embedding, $d(m)$ is the dither value corresponding watermark bit m, and $d(0) = -\Delta_m/4$ and $d(1) = \Delta_m/4$.
3. Finally, the embedding distortion in R is equally distributed over all coefficients. The watermarked coefficients E_i^m are given by
$$E_i^m = E_i + (R^m - R)w_i, \quad 1 \le i \le N_m. \quad (5)$$

Extract Algorithm: Extract one bit $\hat{m}$ from L_m coefficients, $\hat{E}_1, \hat{E}_2, \ldots, \hat{E}_{L_m}$ with key k.

1. First, the same pseudorandom antipodal sequence $w_1, w_2, \ldots, w_{N_m}$ is generated by the key k and correlated with image coefficients,
$$\hat{R} = \frac{1}{N_m} \sum_{i=1}^{N_m} w_i \hat{E}_i. \quad (6)$$
2. The watermark bit $\hat{m}$ is extracted from $\hat{R}$ as follows
$$\hat{m} = \left[\frac{\hat{R} - d(0)}{\Delta_m/2} \right] \mod 2 \quad (7)$$
where Δ_m is the quantization step for MAC embedding, $d(0) = -\Delta_m/4$.

Table 1: STDM embedding and extraction algorithms.

2.3. AMAC Generation and Embedding

We also generate a soft authenticator to determine the degree of distortion and its location. Approximate message authentication codes (AMAC) have been proposed which are able to estimate probabilistically the degree of bitwise similarity of two digital messages [3]. The AMAC generation function is basically a majority function which has the following important feature: similar messages are likely to have similar AMACs. We use its insensitivity feature to estimate the degree of possible distortion on a protected image.

In order to localize possible distortion and estimate its degree of severity in individual 8×8 DCT blocks, we generate and embed AMACs in every 8×8 image block independently. The steps described next are carried out independently in each 8×8 block, denoted by $C^m(i,j), 1 \le i, j \le 8$.

2.3.1. MSB Component Extraction and AMAC Generation

Given 64 coefficients in one coefficient block, we generate 2 bits of AMAC and then embed them in the coefficients $C^m(5,5)$ and $C^m(6,6)$. First, we carry out a quantization on coefficients as $MSB(i,j) = [C^m(i,j)/\Delta_a]$ where Δ_a be the quantization step. On those two coefficients at $(5,5)$ and $(6,6)$ for embedding, we take a further MSB-LSB decomposition: $LSB(i,j) = MSB(i,j) \mod 2$, and $MSB(i,j) = \lfloor MSB(i,j)/2 \rfloor$.

We then generate AMAC from the extracted MSB components $MSB(i,j), 1 \le i, j \le 8$. Let L_a be the AMAC length. Choose a positive odd integer N_a such that $L_a \times N_a$ is greater than or equal to the length of the binary sequence of MSB components. The binary sequence of MSB components is padded with zeros, if necessary, to the length $L_a \times N_a$. Next, the padded sequence is permuted using a pseudorandom permutation according to a key (and possibly the block index) to enforce security. The permuted sequence is masked by XORing all of its bit with a pseudorandom binary sequence generated by the key.

A majority calculation is performed on the permuted and masked sequence. First, re-format the sequence into L_a rows and N_a columns. Denote the array by $B(i,j), 1 \le i \le L_a, 1 \le j \le N_a$. For each row i, compute the majority bit, i.e., the bit which occurs most frequently in the row, as follows,
$$b_i = \begin{cases} 1 & \text{if } \sum_{j=1}^{N_a} B(i,j) > \frac{N_a}{2} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$
for $i = 1, 2, \cdots, L_a$. Since N_a is odd, there is no ambiguity to have a majority bit. The majority bits $(b_1, b_2, \ldots, b_{L_a})$ constitute the generated AMAC sequence.

2.3.2. AMAC Embedding in LSB Component

The generated AMAC bits $(b_1, b_2, \ldots, b_{L_a})$ are embedded in the LSB components of the coefficients at $(5,5)$ and $(6,6)$. Suppose $b_l, 1 \le l \le L_a$ is embedded in coefficient $C^m(i,j)$. The embedding function is as follows,
$$C^a(i,j) = \big(MSB(i,j) * 2 + b_l\big)\Delta_a \quad (9)$$
where Δ_a is the quantization step for AMAC embedding.

2.4. Authentication Verification

To verify authenticity of a test image, the same MAC and AMAC generation algorithms are performed. The embedded MAC and AMAC watermarks are extracted and compared with the respective MAC and AMAC authenticators to verify whether the test image is maliciously or incidentally modified, whether the modification is significant or not, and where

the modification is. The watermark extraction algorithm is basically the "inverse" operation of the embedding one.

The bit error rate (BER) between the generated MAC $(m_1^s, m_2^s, \ldots, m_{L_m}^s)$ and the extracted MAC watermark $(m_1^w, m_2^w, \ldots, m_{L_m}^w)$ is given by

$$\mathrm{BER}_M = \frac{1}{L_m} \sum_{i=1}^{L_m} m_i^s \oplus m_i^w \tag{10}$$

where $\oplus$ denotes exclusive-or operation. The BER of the MAC authenticator is used for authentication judgement whether the test image has been modified maliciously or incidentally. When the content feature is modified, the output of the MAC will be totally different, which leads to a BER around $1/2$. However, when only the embedded watermark is modified, the BER will be small. Thus, $\mathrm{BER}_M = 0$ represents no modification, $\mathrm{BER}_M < T_1$, for some threshold $0 < T_1 < 0.5$, represents corruption only on the embedded watermark, and $\mathrm{BER}_M > T_1$ represents corruption on the content feature, which constitutes a malicious attack.

The bit error rate (BER) of the generated AMAC sequence $(b_1^s, b_2^s, \ldots, b_{L_a}^s)$ and the extracted AMAC watermark $(b_1^w, b_2^w, \ldots, b_{L_a}^w)$ in block (u, v) is given by

$$A_{u,v} = \frac{1}{L_a} \sum_{l=1}^{L_a} b_l^s \oplus b_l^w. \tag{11}$$

The BER $A_{u,v}$ tells the degree of distortion locally in block (u, v). Since the AMAC extraction algorithm is repeated in all blocks, so we obtain a matrix $A_{u,v}, 1 \le u \le \lceil M/8 \rceil, 1 \le v \le \lceil N/8 \rceil$. This AMAC BER matrix shows a distribution of distortion in the entire image. When only part of the image is tampered, the matrix $A_{u,v}$ can locate the tampered area by showing its nonzero elements. The overall BER of the AMAC in the entire image is defined as the average over all blocks,

$$\mathrm{BER}_A = \frac{1}{\lceil M/8 \rceil \lceil N/8 \rceil} \sum_{u=1}^{\lceil M/8 \rceil} \sum_{v=1}^{\lceil N/8 \rceil} A_{u,v} \tag{12}$$

In general, small value of BER_A stands for slight modification on the image while large BER_A around 0.5 tells severe distortion on the image.

Combining the BERs of the MAC and AMAC, we can distinguish different types of distortions. Different authentication decisions can be made from the detection values BER_M and BER_A (or $A_{u,v}$), which are summarized in Table 2.

3. SIMULATION RESULTS

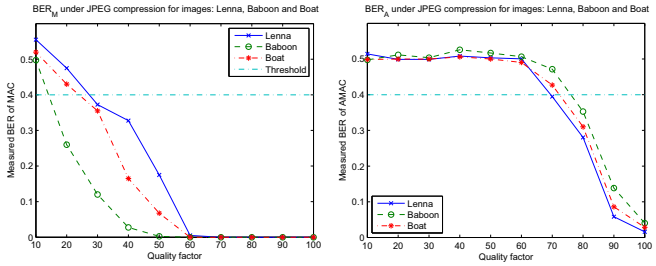
We test the proposed algorithm on real images. Three real images are tested: Lenna, Baboon and Boat. These images are first authenticated using a key to generate the watermarked images. Different incidental and malicious distortions are applied to the watermarked images and the detection values of the MAC and AMAC authenticators are calculated. Decision on the distortion type from the calculated detection values is compared with the actual distortion type to show semi-fragility of the proposed algorithm.

Decision	$\mathrm{BER}_A < T_2$	$\mathrm{BER}_A > T_2$
$\mathrm{BER}_M < T_1$	Image content unchanged. Minor incidental distortion	Image content unchanged. Major incidental distortion.
$\mathrm{BER}_M > T_1$	Minor malicious distortion	Severe malicious distortion

Table 2: Authentication decisions from bit error rates of the MAC and AMAC, BER_M and BER_A, where $0 \le T_1, T_2 \le 0.5$ are two given thresholds.

3.1. Robustness to Incidental Noises

First, we test JPEG compression on all three test images. Authentication detection values, e.g. BERs of MAC and AMAC bits are calculated from the watermarked images after JPEG compression with various quality factors. The BERs of MAC and AMAC bits are shown in Fig. 2. The proposed scheme is robust to JPEG compression with quality factor down to 30 if the threshold T_1 is set to be 0.4.



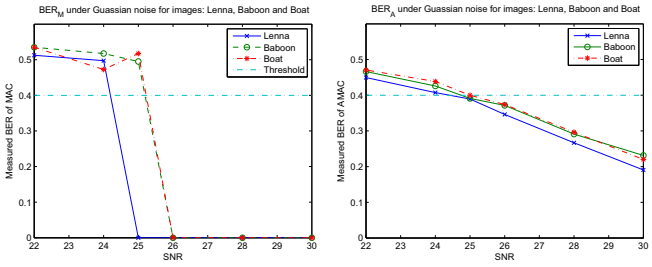
(a) BER of MAC (b) BER of AMAC

Fig. 2: Authentication detection values under JPEG compression for test images.

The authentication detection values under Gaussian noise attacks are plotted in Fig. 3 as the signal-to-noise ratio (SNR) of the additive Gaussian noise varies. Our scheme is robust to additive Gaussian noise with SNR down to $26\,\mathrm{dB}$, or equivalently with noise variance σ_N^2 up to 43.6.

3.2. Fragility against Malicious Tampering

To show fragility against malicious tampering taking place even in a small area, we consider the following four malicious tampering attacks: (A) minor tampering on a block of 8×8 pixels, (B) mild tampering on a block of 32×32 pixels, (C) severe tampering on a block 128×128 pixels, and (D) whole



(a) BER of MAC (b) BER of AMAC

Fig. 3: Authentication detection values under Gaussian noise for test images.

image replacement on entire 512×512 pixels. The location of any malicious tampering is assumed to be random in the watermarked image. In our experiment, when an image block is tampered, the tampered block is replaced by a random block of the same size from a different image.

Image	Malicious tampering			
	A: Minor	B: Mild	C: Severe	D: Whole
Lenna	76	100	100	100
Baboon	75	100	100	100
Boat	73	100	100	100

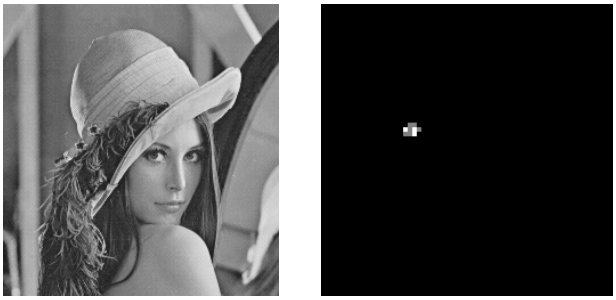
Table 3: The number of successful detection in 100 runs for four types of tampering.

We test these four types of tampering on three test images: Lenna, Baboon, and Boat. Each type of tampering runs 100 times. Each time the measured BER of MAC bits is compared to the threshold $T_1 = 0.4$ and the number of successful detection of malicious tampering is counted and is shown in the Table 3. We see that tampering is detected 100% except when it happens in a very small block of 8×8 pixels.

Image	Malicious tampering			
	A: Minor	B: Mild	C: Severe	D: Whole
Lenna	0.0004	0.0029	0.0370	0.5020
Baboon	0.0005	0.0028	0.0387	0.4998
Boat	0.0005	0.0033	0.0344	0.5015

Table 4: Measured BER of AMAC bits under 4 different tampering attacks.

Table 4 shows the measured BER of AMAC bits under 4 different tampering attacks for three test images. The measured BER of AMAC bits approximately reflects how many pixels are affected in the entire image. The location of malicious tampering can be identified by displaying the bit error matrix $A_{u,v}$ for all blocks. Fig. 4b is an example of $A_{u,v}$ calculated from the tampered image Lenna shown in Fig. 4a, in



(a) Tampered Lenna. (b) BER matrix $A_{u,v}$.

Fig. 4: Malicious tampering of an additional flower on the hat is located by displaying the matrix $A_{u,v}, 1 \leq u \leq \lceil M/8 \rceil, 1 \leq v \leq \lceil N/8 \rceil$.

which an additional flower is maliciously placed on the girl's hat.

4. CONCLUSION

This paper proposes and implements a practical secure semi-fragile scheme for natural images. The most important feature of our proposed algorithm is security against malicious tampering even if the attacker knows everything about the algorithm except the secret key. Another important feature of our system is that our system is able to differentiate malicious tampering from incidental distortions even if the malicious tampering modifies only a small portion of the protected image. Furthermore, our scheme has the ability to determine the degree of the distortion and its location in the entire image.

5. REFERENCES

[1] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing JPEG compression from malicious manipulation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 153–168, Feb. 2001.

[2] Y. Zhao, P. Campisi, and D. Kundur, "Dual domain watermarking for authentication and compression of cultural heritage images," *IEEE Trans. Image Processing*, vol. 13, no. 2, Feb. 2004.

[3] L. Xie, G. R. Arce, and R. F. Gravman, "Approximate image message authentication codes," *IEEE Trans. Multimedia*, vol. 3, no. 2, pp. 242–252, June 1998.

[4] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.