A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation

Ching-Yung Lin, *Member, IEEE,* and Shih-Fu Chang, *Member, IEEE*

***Abstract*—Image authentication verifies the originality of an image by detecting malicious manipulations. Its goal is different from that of image watermarking, which embeds into the image a signature surviving most manipulations. Most existing methods for image authentication treat all types of manipulation equally (i.e., as unacceptable). However, some practical applications demand techniques that can distinguish acceptable manipulations (e.g., compression) from malicious ones. In this paper, we present an effective technique for image authentication which can prevent malicious manipulations but allow JPEG lossy compression. The authentication signature is based on the invariance of the relationships between discrete cosine transform (DCT) coefficients at the same position in separate blocks of an image. These relationships are preserved when DCT coefficients are quantized in JPEG compression. Our proposed method can distinguish malicious manipulations from JPEG lossy compression regardless of the compression ratio or the number of compression iterations. We describe adaptive methods with probabilistic guarantee to handle distortions introduced by various acceptable manipulations such as integer rounding, image filtering, image enhancement, or scaling-recaling. We also present theoretical and experimental results to demonstrate the effectiveness of the technique.**

***Index Terms*—Authentication, digital signature, integrity, JPEG, manipulation, security.**

I. INTRODUCTION

THE WELL-KNOWN adage that “seeing is believing” is no longer true due to the availability of powerful image manipulation software. This technical development has decreased the credibility that photography used to achieve.

Development of robust image authentication techniques becomes an important issue. If we consider a digital image to be merely an ordinary bitstream on which no modification is allowed, then there is not much difference between image authentication and other message authentication problems. Two methods have been suggested for achieving the authenticity of digital images: having a digital camera sign the image using a digital signature [8], or embedding a secret code in the image [26]. The first method uses an encrypted digital “signature,” which is generated in the capturing devices. A digital signature is based on the method of Public Key Encryption [5], [22]. A private key is used to encrypt a hashed version of the image. This encrypted message is called the “signature” of the image, and it provides a way to ensure that this signature cannot be forged. This signature then travels with the image. The authentication process of this image needs an associated public key to decrypt the signature. The image received for authentication is hashed and compared to the codes of the signature. If they match, then the received image is authenticated. The second method embeds a “watermark” in an image [15], [26], [27]. The fragile watermark will usually be destroyed after manipulation. Authenticity is determined by examining the watermark extracted from the received image. Both the above methods have clear drawbacks. Authenticity will not be preserved unless every pixel of the images is unchanged. However, since lossy compression such as JPEG is often acceptable—or even desired—in practical applications, an authentication method needs to be able to distinguish lossy compression from malicious manipulations.

Manipulations on images can be considered in two ways: *method* and *purpose*. Manipulation methods include compression, format, transformation, shifting, scaling, cropping, quantization, filtering, replacement, *etc*. The purpose of manipulations may be *transformation* or *attack*. The former are usually acceptable, and the latter unacceptable. We list two kinds of transformation of representation below.

1) Format transformation and lossless compression. Disregarding the noise caused by the precision limitation during computation, pixel values are not changed after these manipulations. Therefore, we exclude these manipulations in the discussion in this paper.
2) Application-specific transformations. Some applications may require the lossy compression in order to satisfy the resource constraints on bandwidth or storage. Some applications may also need to enhance the image quality, crop the image, change the size, or perform some other operations. A common aspect of these manipulations is that they change the pixel values, which results in different levels of visual distortion in the image. Usually, most of these operations try to minimize the visual distortion.

Attacks, or malicious manipulations, change the image to a new one which carries a different visual meaning to the observer. One typical example is replacing some parts of the image with different content.

It is difficult for an authenticator to know the purpose of manipulation. A practical approach is to design an authenticator based on the manipulation method. In this paper, we design an authenticator which accepts format transformation, lossless compression, and the popular JPEG lossy compression. The authenticator rejects replacement manipulations because they are

Manuscript received September 11, 1998; revised October 3, 2000. This paper was recommended by Associate Editor R. Koenen.

The authors are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA.

Publisher Item Identifier S 1051-8215(01)01246-0.

frequently used for attacks. Our authenticator does not aim to reject or accept, in absolute terms, other manipulation methods because the problem of whether they are acceptable depends on applications. But, if necessary, some manipulations can be clearly specified by users, such as shifting, cropping, or constant intensity enhancement. We will discuss this more rigorously later. The proposed authentication techniques has been extended and applied to MPEG video authentication as well [16].

For an image, there are some invariance properties which can be preserved during JPEG lossy compression. Let us consider the relationship between two DCT coefficients of the same position in two separate 8×8 blocks of an image. This relationship will hold even if these coefficients are quantized by an arbitrary quantization table in a JPEG compression process. In this paper, we will use this invariance property and propose a robust authentication method which can distinguish malicious manipulations from JPEG lossy compression.

A comprehensible list of multimedia authentication research papers can be found in [17]. Bhattacha and Kutter proposed an authentication method which extracts "salient" image feature points by using a scale interaction model and Mexican-Hat wavelets [1]. They generate a digital signature based on the locations of these feature points. The advantage of this technique is its compact signature length. But, the selection process and relevance of the selected points are not clear. This technique may not be adequate for detecting some crop-and-replace manipulations inside the objects. Its robustness to lossy compression is also unclear. Queluz proposed techniques to generate digital signatures based on moments and edges [20]. Moment features ignore the spatial distribution of pixels. Images can be easily manipulated without changing their moments. Edge-based features may be a good choice for image authentication because the contour of objects should keep consistent for acceptable manipulations. However, several issues have to be further solved such as the reduction of signature length, the consistency of edge detector, and the robustness to color manipulations. Fridrich proposed a robust watermarking technique for authentication [6], [7]. He divided images into 64×64 blocks. For each block, quasi-VQ codes are embedded by the spread spectrum method [3]. This technique is robust to manipulations. But, it cannot detect small area modification. The error between the extracted watermark and the reconstructed quasi-VQ codes is too large after JPEG compression [7]. Therefore, this technique would be hard to distinguish malicious manipulations from JPEG compressions.

This paper is organized as follows. We briefly review the JPEG system in Section II. In Section III, a general system for authentication will be proposed. Also, we will describe how to control parameters for different practical uses. A simple example is shown in this section. We will present rigorous performance analysis in Section IV. Experimental results will be shown in Section V. In Section VI, we will present conclusions and discuss future work.

II. Review of JPEG Lossy Compression

In this section, we briefly review the JPEG lossy compression standard. At the input to the JPEG [25] encoder, the source image, X, is grouped into $\wp$ nonoverlapping 8×8 blocks, $X = \bigcup_{p=1}^{\wp} \mathbf{X_p}$. Each block is sent sequentially to the discrete cosine transform (DCT). Instead of representing each block, $\mathbf{X_p}$, as a 8×8 matrix, we can rewrite it as a 64×1 vector following the "zigzag" order [25]. Therefore, the DCT coefficients, $\mathbf{F_p}$, of the vector, $\mathbf{X_p}$, can be considered as a linear transformation of $\mathbf{X_p}$ with a 64×64 transformation matrix $\mathbf{D}$, *s.t.*,

$$\mathbf{F_p} = \mathbf{D}\mathbf{X_p}. \tag{1}$$

Each of the 64 DCT coefficients is uniformly quantized with a 64-element quantization table $\mathbf{Q}$. In JPEG, the same table is used on all blocks of an image. (For color images, there could be three quantization tables for YUV domains, respectively.) Quantization is defined as the division of each DCT coefficient by its corresponding quantizer step size, and rounding to the nearest integer:

$$\tilde{\mathbf{f}}_\mathbf{p}(\nu) \equiv \text{Integer Round}\left(\frac{\mathbf{F_p}(\nu)}{\mathbf{Q}(\nu)}\right) \tag{2}$$

where $\nu = 1, \ldots, 64$. In . (2), $\tilde{\mathbf{f}}_\mathbf{p}$ is the output of the quantizer. We define $\tilde{\mathbf{F}}_\mathbf{p}$, a quantized approximation of $\mathbf{F_p}$, as

$$\tilde{\mathbf{F}}_\mathbf{p}(\nu) \equiv \tilde{\mathbf{f}}_\mathbf{p}(\nu) \cdot \mathbf{Q}(\nu). \tag{3}$$

In addition to quantization, JPEG also includes scan order conversion, dc differential encoding, and entropy coding. Inverse DCT (IDCT) is used to convert $\tilde{\mathbf{F}}_\mathbf{p}$ to the spatial-domain image block $\tilde{\mathbf{X}}_\mathbf{p}$

$$\tilde{\mathbf{X}}_\mathbf{p} = \mathbf{D}^{-1}\tilde{\mathbf{F}}_\mathbf{p}. \tag{4}$$

All blocks are then tiled to form a decoded image frame.

Theoretically, the results of IDCT are real numbers. However, the brightness of an image is usually represented by an 8-bit integer from 0 to 255, and thus a rounding process mapping those real numbers to integers is necessary. We found that popular JPEG softwares, such as PhotoShop, xv, etc., use the integer-rounding functions in several steps of their DCT and IDCT operators in order to save computation or memory. The input and output of their DCT and IDCT operators are all integers. This approximation may not introduce too much visual distortion but may affect the authentication system performance that we will discuss in more detail in Section IV.

III. Authentication System

The proposed authentication method is shown in Fig. 1. Our method uses a concept similar to that of the digital signature method proposed by Friedman [8], but their technique doesn't survive lossy compression. A signature and an image are generated at the same time. The signature is an encrypted form of the feature codes or hashes of the image. When a user needs to authenticate the image he receives, he should decrypt this signature and compare the feature codes (or hashed values) of this image to their corresponding values in the original signature. If they match, this image is said to be "authenticated." The most important difference between our method and Friedman's "trustworthy camera" is that we use invariance properties in

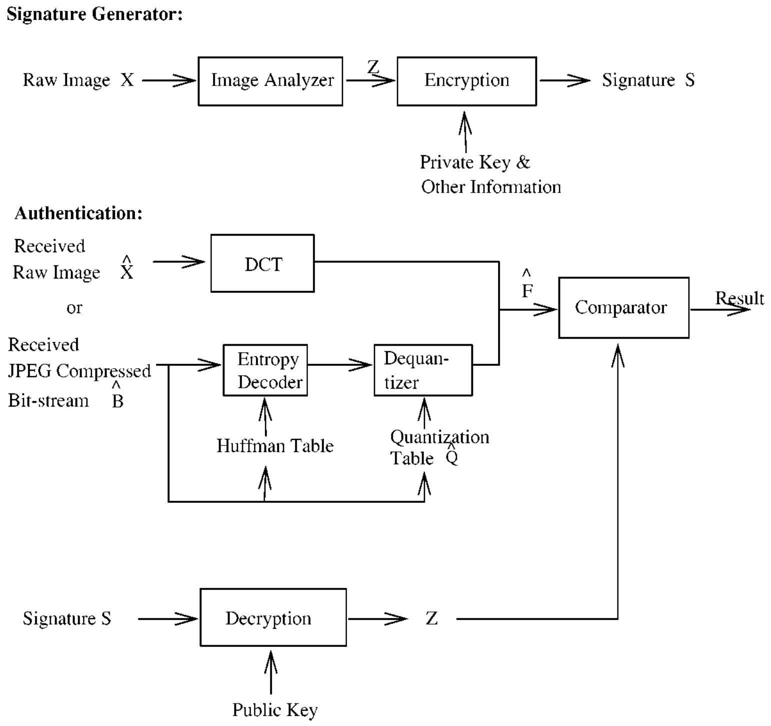


Fig. 1. Signature generator and authentication process.

JPEG lossy compression as robust feature codes instead of using hashes of raw images.

A. *Invariants of an Image Before and After JPEG Compression*

From the compression process of JPEG, we have found that some quantitative invariants and predictable properties can be extracted. Two steps in the JPEG compression process reduce the required bits representing an image: 1) *quantization and rounding of the DCT coefficients* and 2) *entropy coding*. The second step is a lossless operation. The first step is a lossy operation which alters pixel values but keeps important visual characteristics of the image. Therefore, if robust feature codes are expected for authentication, they must survive this step. The following theorems provide a technical basis for generating such robust feature codes. Proofs of these theorems are included in the Appendix.

Theorem 1: Assume $\mathbf{F_p}$ and $\mathbf{F_q}$ are DCT coefficient vectors of two arbitrary 8×8 nonoverlapping blocks of image X, and $\mathbf{Q}$ is the quantization table of JPEG lossy compression. $\forall \nu \in [1, \ldots, 64]$ and $\mathbf{p}, \mathbf{q} \in [1, \ldots, \wp]$, where $\wp$ is the total number of blocks, define $\mathbf{\Delta F_{p,q}} \equiv \mathbf{F_p} - \mathbf{F_q}$ and $\mathbf{\Delta \tilde{F}_{p,q}} \equiv \mathbf{\tilde{F}_p} - \mathbf{\tilde{F}_q}$ where $\mathbf{\tilde{F}_p}$ is defined as $\mathbf{\tilde{F}_p}(\nu) \equiv \text{Integer Round}(\mathbf{F_p}(\nu)/\mathbf{Q}(\nu)) \cdot \mathbf{Q}(\nu)$. Then, the following properties must be true:

1) if $\mathbf{\Delta F_{p,q}}(\nu) > 0$, then $\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \geq 0$;
2) else if $\mathbf{\Delta F_{p,q}}(\nu) < 0$, then $\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \leq 0$;
3) else $\mathbf{\Delta F_{p,q}}(\nu) = 0$, then $\mathbf{\Delta \tilde{F}_{p,q}}(\nu) = 0$.

□

In summary, because all DCT coefficient matrices are divided by the same quantization table in the JPEG compression process, the relationship between two DCT coefficients of the same coordinate position will not change after quantization. The only exception is that "*greater than*" or "*less than*" may become "*equal*" due to the rounding effect of quantization. The above theorem assumes that the same quantization table is used for the whole image. Theorem 1 is valid no matter how many recompression iterations and what the quantization tables are used.

For practical implementations, the quantization table can be extracted from the compressed file or estimated from the DCT coefficients of decompressed file. Note that Theorem 1 only preserve the sign of coefficient differences. The following theorem extends it to preserve the difference values, with various resolutions.

Theorem 2: Use the parameters defined in Theorem 1. Assume a fixed threshold $k \in \Re$. $\forall \nu$, define $\tilde{k}_\nu \equiv \text{Integer Round}(k/\mathbf{Q}(\nu))$. Then, if $\mathbf{\Delta F_{p,q}}(\nu) > k$

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \geq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \dfrac{k}{\mathbf{Q}(\nu)} \in Z \\ \left(\tilde{k}_\nu - 1\right) \cdot \mathbf{Q}(\nu), & \text{elsewhere} \end{cases} \tag{5}$$

else if $\mathbf{\Delta F_{p,q}}(\nu) < k$,

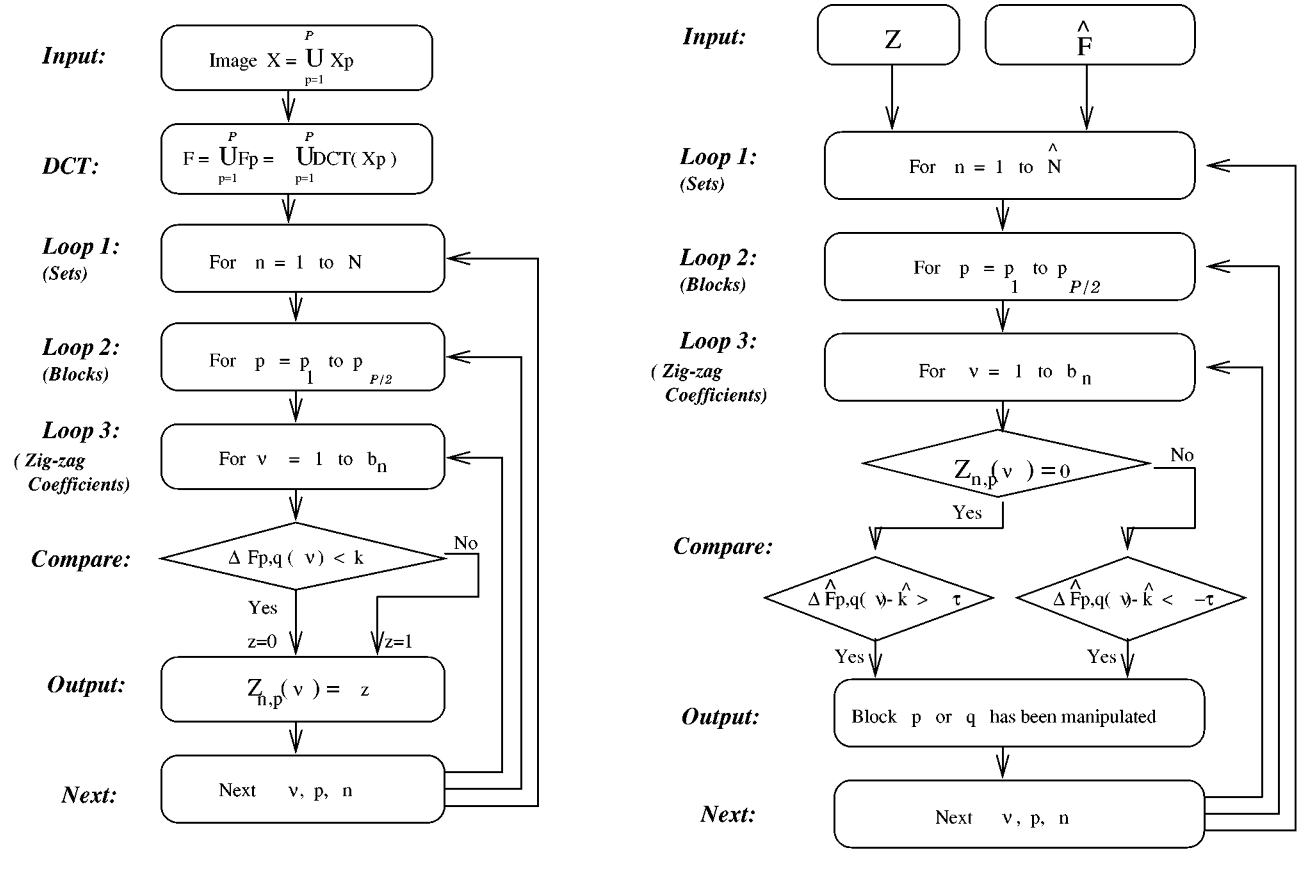
$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) \leq \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \dfrac{k}{\mathbf{Q}(\nu)} \in Z \\ \left(\tilde{k}_\nu + 1\right) \cdot \mathbf{Q}(\nu), & \text{elsewhere} \end{cases} \tag{6}$$

else $\mathbf{\Delta F_{p,q}}(\nu) = k$

$$\mathbf{\Delta \tilde{F}_{p,q}}(\nu) = \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \dfrac{k}{\mathbf{Q}(\nu)} \in Z \\ \left(\tilde{k}_\nu \text{ or } \tilde{k}_\nu \pm 1\right) \cdot \mathbf{Q}(\nu), & \text{elsewhere.} \end{cases} \tag{7}$$

□

In Theorem 2, k is a designated threshold value used to bound the difference of two DCT coefficients of the same position in



(a)

(b)

Fig. 2. (a) Feature extraction process. (b) Authentication comparator.

two separate blocks of an image. In contrast, Theorem 1 only describes the invariance property of the sign of $\mathbf{\Delta F_{p,q}}$. We can consider Theorem 1 as a special case of Theorem 2 (with k set to be 0). Several different k's (e.g., a series of binary division of a fixed dynamic range) can be used for a single authentication system of different levels of strength. Based on Theorem 2, we can predict the difference relationships between coefficients after compression. Extension of the invariance property to the case of variable quantization table is included in Appendix I.

As shown in Fig. 1, by applying Theorem 1 and Theorem 2, we can extract feature codes Z of an image from the relationships between two DCT coefficients of the same position in two separate blocks. These feature codes are then encrypted as a signature. For the authentication process, a user has to calculate the DCT coefficients of the image, and compare them to the features decrypted from the digital signature S. This image is said to be authenticated if all the DCT coefficient relationships satisfy the criteria predicted by the features of the original image.

B. Image Analyzer: Feature Extraction

Fig. 2(a) is the flowchart of the feature extraction process. First, a digital image X is sent into the image analyzer. Each 8×8 block of this image is then transformed to the DCT coefficients.

There are three loops for generating feature codes:

1) *Loop 1:* Generate N sets of feature codes, $Z_{n,p}$, $n = 1$ to N. Each set uses different k and b_n, where k is defined in Theorem 2, b_n is the number of DCT coefficients compared in each block pair.
2) *Loop 2:* Iterate over all possible block pairs, $p = p_1$ to $p_{\wp/2}$, where $\wp$ is the total number of blocks in the image.
3) *Loop 3:* Iterate over each of the b_n selected coefficient pairs.

In Loop 1, N sets of feature codes are generated. For each set, parameter b_n represents how many bits are generated in each block. Parameter k represents the precision threshold used in Theorem 2. The first set $k = 0$ protects the sign of $\Delta F_{p,\,q}$. From the second set to the last set, k's are set to protect the magnitude of $\Delta F_{p,\,q}$ with increasing accuracy. We will discuss how to define the thresholds later in this section.

In Loop 2, we need to form DCT blocks into pairs. As defined in Theorem 2, the DCT coefficient difference between block p and block q is computed. Let us denote one set of blocks $P_p = \{p_1, p_2, \ldots, p_{\wp/2}\}$ and another set of blocks $P_q = \{q_1, q_2, \ldots, q_{\wp/2}\}$. For example, P_p can be all the even blocks, $\{0, 2, 4, \ldots, \wp - 1\}$, and P_q can be all the odd blocks, $\{1, 3, 5, \ldots, \wp - 2\}$. The formation of all blocks in an image into pairs can be based on an arbitrary mapping function W, as long as the following conditions are kept:

$$P_q = W(P_p) \tag{8}$$

and

$$P_p \cap P_q = \varnothing, \qquad P_p \cup P_q = P. \tag{9}$$

If redundancy is allowed, P_p and P_q each may contain more blocks than $\wp/2$. The choice of the mapping function W can serve as a secret parameter, used to enhance the security of authenticator. For example, the image analyzer uses a seed to generate the mapping function W and provides the seed with the feature codes to the authenticator. Each authenticator can transform this seed into the mapping function. This transformation method will not be published. Therefore, each manufacturer of the image authenticator can implement his/her own transformation method.

In Loop 3, for each block, we compare the b_n selected values (indexed in the zigzag order) in the DCT domain. Both dc and ac values in the DCT domain are used. At first, the difference of dc values in block p and q, $\mathbf{\Delta F_{p,q}}(1)$, is used for comparison. If this value is smaller than k, then a feature code bit $z = 0$ is added to the end of the previous feature code. Otherwise, if this value is greater or equal to k, we will assign $z = 1$. (We classify two cases, "greater" and "equal," to the same type because the probability of $\mathbf{\Delta F_{p,q}}(\nu) = 0$ is quite small. If they are classified into three different types, i.e., "greater," "equal," and "less than," two bits should be used in this case. This will result in the increased length of feature codes.) Thereafter, the differences in selected ac values are compared with k. Only $b_n - 1$ ac differences, are used in this process. After Loops 1, 2, and 3 are completed, the feature codes Z of this image are generated. Usually, the b_n selected positions are located in the low and middle frequency bands for the following two reasons: 1) they are usually larger than the high-band coefficients because of energy concentration and 2) their values are usually conserved after JPEG compression because the values in the quantization table $\mathbf{Q}$ in these bands are small.

1) Precision Thresholds and Other Considerations: Theoretically, threshold values k can be determined arbitrarily, and they may vary for different n and ν. In our system, for the first set, all $k_{1,p}(\nu)$ are set to zeros. We use a binary division method to set thresholds for other sets. Assume the dynamic range of $\mathbf{\Delta F_{p,q}}(\nu)$ is from $-\zeta$ to ζ. If we know that $\mathbf{\Delta F_{p,q}}(\nu) < 0$ in the first set, then we can set the threshold in the second set as $-\zeta/2$. Furthermore, if we know that this value $\mathbf{\Delta F_{p,q}}(\nu) > -\zeta/2$ in the second set, the threshold in the third set can be set as $-\zeta/4$. These thresholds result in dynamic binary decision ranges. This method protects the magnitude of $\mathbf{\Delta F_{p,q}}(\nu)$ with an increasing accuracy as more sets are being used. The larger N is, the more precisely will the coefficient differences be limited.

Define a constant ζ which is a power of 2, and the threshold used in the nth set of block p at the position ν is $k_{n,p}(\nu)$. A closed form of $k_{n,p}(\nu)$ is

$$k_{n,p}(\nu) = \zeta \sum_{i=1}^{n-1} \left(\frac{1}{2}\right)^{i} (-1)^{Z_{i,p}(\nu)+1}, \qquad n > 1. \tag{10}$$

To simplify the notation in later discussions, we use $k = k_{n,p}(\nu)$ instead.

In addition to the parameters used in the three loops, some extra information about the image is necessary for defeating attacks. In our authentication system, a possible attack is to make a constant change to DCT coefficients at the same location in all blocks. This will not change the difference values between pairs of DCT coefficients from two different blocks. For instance, raising the image intensity uniformly changes the dc parameter in all blocks and defeats the previous approach. To defeat this attack, we record the mean value of DCT coefficients in each (selected) position for all blocks in the signature. These additional feature codes need no more than 64 bytes. When the DCT coefficients are changed by constant values, they will be easily detected by the deviation of their mean values.

C. Authentication Process

Fig. 1 includes the authentication process. It is composed of three parts. First, the received image $\hat{X}$ or $\hat{B}$, has to be transformed to the DCT domain, $\hat{F}$. This involves the DCT transform block by block if a raw image $\hat{X}$ is used. If a JPEG compressed image $\hat{B}$ is used, a parser has to be used for reconstructing the Huffman Table and Quantization Table $\hat{\mathbf{Q}}$. The signature S has to be decrypted to reconstruct feature codes Z. After $\hat{F}$ and Z are available, they will be sent to the authentication comparator in order to determine whether this image has been manipulated.

The Authentication Comparator is shown in Fig. 2(b). Similar to the three loops in the image analyzer, there are also three corresponding loops here. In Loop 1, the number of loops n can be different from the one used in the Image Analyzer. Fewer loops may be used. Loop 2 and Loop 3 are the same as those used in the Image Analyzer. Inside these loops, we have to compare each of the DCT coefficient relationships obtained from the original image and that of the image received.

From Theorem 2, we can define

$$\hat{k} = \begin{cases} \tilde{k}_\nu \cdot \mathbf{Q}(\nu), & \dfrac{k}{\mathbf{Q}(\nu)} \text{ is an integer} \\ \left(\tilde{k}_\nu + 1\right) \cdot \mathbf{Q}(\nu), & \dfrac{k}{\mathbf{Q}(\nu)} \text{ is not an integer} \\ & \text{and } Z_n(\nu) = 0 \\ \left(\tilde{k}_\nu - 1\right) \cdot \mathbf{Q}(\nu), & \dfrac{k}{\mathbf{Q}(\nu)} \text{ is not an integer} \\ & \text{and } Z_n(\nu) = 1. \end{cases} \tag{11}$$

(*Note that $\hat{k}$ is a function of ν, p, and n.*) Observe from Fig. 2(b), if $Z_n(\nu) = 0$, that is, $\mathbf{\Delta F_{p,q}}(\nu) < k$, then $\mathbf{\Delta \hat{F}_{p,q}}(\nu) - \hat{k} \leq 0$ must be satisfied. Therefore, if $\mathbf{\Delta \hat{F}_{p,q}}(\nu) - \hat{k} > 0$, we know that some parameters of block p or q must have been modified. Similar results can be obtained in the case of $\mathbf{\Delta F_{p,q}}(\nu) \geq k$.

However, some integer rounding noise may be introduced if the following cases occur: 1) the image is converted back to integral pixel values during the decode–reencode process; 2) the compressor and the signature generator use different chromatic decimation algorithms for color images; or 3) the JPEG encoder calculates imprecise DCT. Therefore, we must introduce a tolerance bound τ in the authenticator. We augment the comparing process with the following position.

Proposition 1: Block p or q can be said to be manipulated if

$$\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) - \hat{k} > \tau \tag{12}$$

for the case of $\mathbf{\Delta F}_{\mathbf{p},\mathbf{q}}(\nu) - k < 0$ [or equivalently $Z_n(\nu) = 0$], or if

$$\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) - \hat{k} < -\tau \tag{13}$$

for the case of $\mathbf{\Delta F}_{\mathbf{p},\mathbf{q}}(\nu) - k \geq 0$ [or equivalently $Z_n(\nu) = 1$].

The tolerance τ is determined by the level of integer rounding errors. Optimal levels of the rounding tolerance will be discussed in Section IV-A.

Note that the result of authentication can be a binary indicator *true or false* for the whole image, or it may indicate the authenticity or forgery of specific parts in an image.

1) Other Considerations: Manipulation in specific block pairs can be located by the proposed technique. However, the authenticator using nonoverlapping sets in (9) will not be able to identify which block in the pair has been modified. If identification of specific blocks is needed, we can use overlapping sets in (9). Identifying local changes is very useful to some applications in which both global and local contents are important. For instance, in a picture of ten people, even if a man's face has been substituted by that of another person or has been removed, another part of the image can still be verified to authenticate the appearance of the other nine people. Another advantage is that the system can verify authenticity in a selected area (e.g., some news agency may cut out boundary areas of file photos).

Boundary cropping and/or position shifting are often performed on images to suit application needs. The proposed authentication signature is sensitive to cropping and shifting. However, for cropping, image block pairs that are not affected by cropping may still be authenticated. If cropping is allowed in some situations, we can design a robust digital signature with carefully selected mapping function, e.g., selecting pairs from adjacent blocks. For shifting, if no DCT quantization is done on the shifted image, e.g., shifting in the pixel domain only), the shifted image can be adjusted to the right position that results in the matched DCT block structure. Then, the DCT domain signature can be verified.

Constant intensity change in the image is sometimes expected, especially when the image is too dark or too bright. Our proposed authenticator solves this problem by relaxing the change threshold $\tau_s(1)$ of the mean value of dc coefficients.

Scaling is a common operation in many situations. For instance, a user may scan a picture with high resolution, and then down-sample it to an appropriate size. In this case, the signature generator has to record the original size of the image. Then, the authenticator can resize the image to its original size before the authentication process. Because the distribution of these sampling/interpolation noises can be modeled by a Gaussian function whose variance is not too large [11], there will be no large changes in the DCT coefficients. Similar to the recompression distortions, these changes can be accepted by setting adequate tolerance values.

Other lossy compression techniques such as wavelet-based methods or color–space decimation methods can be also considered as noise-adding processes. Similarly, we can use larger tolerances for these cases. Filtering, such as low-pass filtering and edge enhancement, may cause more visual changes and may cause challenges to the proposed technique. However, if the change in pixel values is not too large, we can consider them as some kind of noise and use adequate tolerance values. This strategy of adjusting tolerance levels can also be applied to other operations as well.

The authenticator is sometimes expected to pass only those images that are compressed by JPEG up to a certain compression ratio or quality factor. For example, if the image is JPEG compressed below the 20 : 1 ratio, the image is acceptable. Otherwise, if it is compressed more, it will fail the test. The argument for failing highly compressed images is that such images usually have poor quality and should not be considered as authentic. To satisfy this need, we can apply one of the following methods. The first one is to calculate the compression ratio from the raw image size and the compressed file size. If it is too high, the authenticator can reject it before any authenticating process. The second method is to calculate the increase of the number of the "equal" signature bits after compression. The number of "equal" signature bits increases if the image is compressed more. We can set a threshold on this change to reject those images that have too many "equal" coefficients in the block pairs.

D. Encryption, Decryption, and Signature Length

The feature codes are encrypted by a secret private key of the Public Key method. As described in Section III-B, the length l_f of feature codes is determined by the comparison bits $\wp/2 \cdot (\sum_{n=1}^{N} b_n)$, the seeds of the block pair mapping function and selected DCT positions, and the DCT mean values (see Section III-B-1). For instance, assume the image size is $320 \times 240 = 76\,800$ (bytes). In a scenario where 10 bits of feature codes are used for each block pair, i.e., $N = 1$ and $b = 10$. Assume both the seeds are 2 bytes long, and 6 DCT coefficient averages are recorded, then the length of feature codes l_f will be $((40 \times 30)/2) \cdot 10 \cdot (1/8) + 2 + 2 + 6 = 760$ (bytes). The signature length can be further reduced with the reduction of the authenticator's effectiveness. We will analyze this tradeoff in detail in Section IV.

The Public Key algorithm is used so that any user can easily access a public key to decrypt the signature. The most famous public key algorithm is Rivest, Shamir, and Adleman (RSA) [12], [23]. The key length of RSA is variable, but the most commonly used length is 512 bits [23], while the message block size must be smaller than the key length. If we choose to divide the feature codes into B-bit blocks, it needs $\lceil l_f \cdot 8 \cdot (1/B) \rceil$ RSA calculations (where $\lceil x \rceil$ denotes the integer ceiling function). Assume the output length of each RSA is l_r, then the signature length will be $\lceil l_f \cdot 8 \cdot (1/B) \rceil \cdot l_r$ bits. For instance, in previous example, if $B = 510$ and $l_r = 511$ are used, then the RSA algorithm has to be run 12 times and the signature length will be 767 bytes. It is about $1/100$ of the original image size.

A problem with Public Key algorithms is the speed. In hardware, the RSA Public Key algorithm is 1000 times slower than the DES Secret Key algorithm. The difference is about 100 times in software [23]. Therefore, if efficiency is critical, we can choose the Secret Key algorithm instead of the Public Key

TABLE I
TWO DCT COEFFICIENT BLOCKS FOR A 16×8 AREA CUT FROM THE IMAGE "LENNA" (RIGHT EYE REGION)

486	91	-66	-91	-17	-1	14	-0
140	41	44	35	-8	-12	-6	-4
43	108	-54	5	16	13	-9	-0
-143	-21	84	34	22	-0	-12	6
9	-18	-2	-32	8	5	5	12
-23	-9	1	-1	-8	1	2	-0
3	10	-14	4	6	-1	-1	-6
-8	-10	14	3	-1	-2	-2	-3

(a)

727	-188	-3	-28	-16	-4	-6	-1
51	-77	22	45	11	1	2	3
31	-52	-73	-8	5	5	10	7
73	40	-21	-7	1	-13	-2	-2
19	12	-21	-17	4	2	2	-1
20	15	-2	-17	-5	2	-0	-1
16	16	13	1	2	6	-2	0
-1	-3	-6	-12	-6	-1	1	3

(b)

TABLE II
DCT COEFFICIENTS IN TABLE I QUANTIZED BY A UNIFORM MATRIX

480	96	-64	-96	-16	0	16	0
144	48	48	32	-16	-16	0	0
48	112	-48	0	16	16	-16	0
-144	-16	80	32	16	0	-16	0
16	-16	0	-32	16	0	0	16
-16	-16	0	0	-16	0	0	0
0	16	-16	0	0	0	0	0
-16	-16	16	0	0	0	0	0

(a)

720	-192	0	-32	-16	0	0	0
48	-80	16	48	16	0	0	0
32	-48	-80	-16	0	0	16	0
80	48	-16	0	0	-16	0	0
16	16	-16	-16	0	0	0	0
16	16	0	-16	0	0	0	0
16	16	16	0	0	0	0	0
0	0	0	-16	0	0	0	0

(b)

algorithm. The drawback is that users have to keep their secret keys safe, and the image can be authenticated by only the few people who own the secret key.

Implementation of a Public Key Infrastructure (PKI) is necessary for practical application. A PKI is the set of security services that enable the use and management of public-key cryptography and certificates, including key, certificate, and policy management [10]. If the image is generated from a hardware device, such as digital camera or scanner, then the manufacturer can serve as a Certification Authority (CA) issuing all user private–public key pairs. Private keys can be either embedded in the hardware device or issued to the driver software while customers register their information. Any end entity can examine the authenticity of images by requesting the public key and the authentication software from the manufacturer. Similarly, if the image is generated by a Content Holder, he/she can ask a private-public key pairs from any CA, which provides both key pairs and authentication software. The details of the PKI and the related standard $X.509$ can be found in [9].

E. Example: A Small 16×8 Image

We will use a small 16×8 image X as an example to illustrate the proposed authentication technique. This image is divided into two 8×8 blocks, from which DCT coefficients are computed. Therefore, $\wp = 2$. Its DCT coefficients are shown in Table I. For simplicity, only integral values of them are shown in the table.

First, let us consider the case of $N = 1$, i.e., only one threshold value $k = 0$ is used for feature code generation. Assume the first ten coefficients in the zigzag order of the two blocks are compared. In this case, the length of the feature codes, Z, will be 10 bits ($b_1 = 10$) $\mathbf{\Delta F_{1,2}}(1) = -241 < 0$. Therefore, the first bit of the feature codes Z is 0. The second coefficients in the zigzag order are: $\mathbf{F_1}(2) = 91$ and $\mathbf{F_2}(2) = -188$, respectively. Since $\mathbf{\Delta F_{1,2}}(2) = 279 > 0$, the second bit of the feature codes is 1. After ten iterations, the feature codes, Z, are: 0 111 100 110.

Consider longer feature codes, we set $N = 4$, $b_1 = 10$, $b_2 = 6$, $b_3 = 3$, and $b_4 = 1$. The reason for a decreasing value of b_n is that the lower frequency coefficients need more protection than the higher frequency ones. The threshold values (k's) are 0, 128, 64, and 32 (in the absolute form). The first 10 bits of Z are the same as the previous case. For the next 6 bits, the first six coefficients are compared again using $|k| = 128$. For example, since $\mathbf{\Delta F_{1,2}}(1) = -241 < -128$, the 11th bit = 0. $\mathbf{\Delta F_{1,2}}(2) = 279 > 128$, so the 12th bit = 1. The final feature codes are: 01 111 001 100 100 010 110. The length of Z is $\sum_{n=1}^{4} b_n = 20$.

Table II shows the DCT coefficients after quantization (i.e., $\tilde{\mathbf{F}}_\mathbf{1}$ and $\tilde{\mathbf{F}}_\mathbf{2}$) with a uniform matrix of 16. This is to simulate the quantization process in JPEG. Using Fig. 2(b), we authenticate the compressed image by comparing $\mathbf{\Delta}\tilde{\mathbf{F}}_{\mathbf{1},\mathbf{2}}$ to the feature codes Z. For instance, $\mathbf{\Delta}\tilde{\mathbf{F}}_{\mathbf{1},\mathbf{2}}(1) = -240 < 0$ and $Z_1(1) = 0$, this value is authenticated to be true. Similar process continues until all feature codes are used. Note if the quantization table is not known to the authenticator, the first set of codes (with $k = 0$) can still be verified.

Consider an example of manipulation. Assume $X(0, 2)$ and $X(0, 3)$ are modified from 72 and 26 to 172 and 126. (X can be obtained from the IDCT of Table I.) Assume we use the same quantization matrix. Repeating the above process, the authenticator will detect the manipulation due to the mismatch of the fourth bit of the feature codes.

TABLE III
PROPERTIES OF DIFFERENT SYSTEM VARIABLES FROM VIEWPOINTS OF DIFFERENT PARTIES

	Image	Mapping Function & Number of Bits in Sets	Manipulation	Rounding Noise
Signature Generator	fixed	selected	random	random
Authenticator	fixed	fixed	random	random
Attacker	fixed	random	fixed	random
System Evaluation	random	random/fixed	random/fixed	random

F. Color Images

In the JPEG standard, color images are considered to be in the YC_bC_r format. Chromatic components (C_b, C_r) are usually down-sampled at the rate of 2 : 1 (horizontal direction only) or 4 : 1 (one-half in both horizontal and vertical directions). To authenticate a color image, we first down-sample the chromatic components with the sampling rate 4 : 1. Then, we generate the feature codes of Y, C_b, C_r in the same way as described earlier. In the authenticator, if the chromatic components are sampled by 2 : 1, they are subsampled again in the other direction in order to obtain 4 : 1 subsampled color components.

IV. Performance Analysis

The image authenticator is a manipulation detector with two types of error involved: *miss* and *false alarm* [21]. "Miss" refers to the situation in which an image is manipulated by unacceptable manipulations but the system reports the image as authentic. "Miss" is also called *Type II error* in Hypotheses Testing. "False alarm" means that the system reports the existence of manipulation while the image is, in fact, not modified by unacceptable manipulations. It is also called a *Type I error*. In our authentication system, the test is based on block pairs. For each block pair, we perform the following test: H_0—*the pixels in the image block pair are not modified, or modified to new values that can be obtained by the JPEG compression processes*, versus H_1—*the pixels in the image block pair are modified to new values that cannot be obtained by any JPEG process*. The test function is defined in Proposition 1. Conceptual illustration of "Miss," "False alarm," and other scenarios are shown in Fig. 3(a)–(g). I represents the original image. R is the set of images obtained by JPEG compression of I. R_n is R augmented with rounding errors in JPEG compression. S is the set of images passing authentication. S_τ is the set of images passing authentication allowing tolerance bounds. (a), (b), and (d) are correct authentication. (c) is a miss. (d) is a false alarm by S but correct authenticated by S_τ. (e) is a correct authentication by S but missed by S_τ. (f) is a false alarm. (g) is a successful detection of manipulation.

The probabilities of miss (P_m) and false alarm (P_f) are estimated by the signature generator and are useful to users of the authenticator. An additional evaluation metric, the probability of success (P_s) can also be used from the attacker's viewpoint. The attacker may try to manipulate the image based on his best knowledge of the authentication technique. Detailed discussion using these metrics will be shown in this section.

Several variables are needed to estimate these probabilities. We can classify variables to three types: *pre-determined values*, *selectable variables*, and *stochastic variables*. The signature generator estimates a list of P_f and P_m based on different quantization tables and tolerances. Based on the quantization table used in the compressed image, the user may choose tolerances τ to satisfy constraints on P_f and P_m. Various properties of system variable from viewpoints of different parties are shown in Table III.

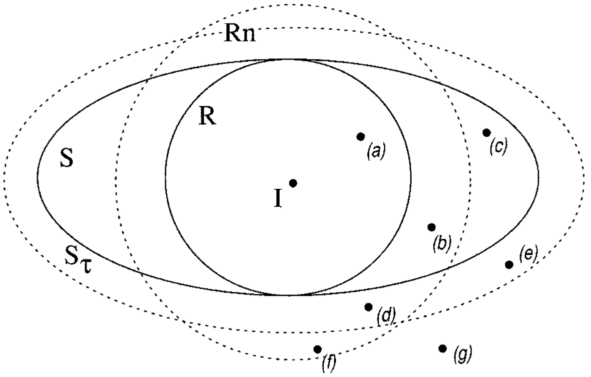


Fig. 3. Conceptual illustration of "miss," "false alarm," and other scenarios.

A. Noise from the Compression Process and the Probability of False Alarm

Rounding noise may be added during the JPEG compression process and may cause false alarm. In practice, computer software and hardware calculate the DCT with finite precision. For some cases, not only the input and the output of DCT operations are integers, but also some of the intermediate values. This will add rounding noise to the DCT values. In addition, some applications may drop small values in the high frequency positions. Combining these considerations, we can modify (2) to

$$\tilde{\mathbf{f}}_{\mathbf{p}}(\nu) = \text{Integer Round}\left(\frac{\mathbf{F}_{\mathbf{P}}(\nu) + N_d}{\mathbf{Q}(\nu)}\right) + N_r \tag{14}$$

where N_d is the noise of DCT operation and N_r is the noise of integer rounding. Both are random variables. N_d usually depends on specific implementations and the number of recompression processes. Also, in most systems, the rounding rules are consistent over different positions and thus the effect of N_r can be ignored in computing DCT coefficient differences.

The probability of false alarm of a block pair (P_f) represents the probability that at least one DCT difference value in the block pair triggers the detector in Proposition 1, because of the effect of rounding noise. We can write P_f as

$$P_f = 1 - \prod_{n=1}^{N} \prod_{\nu=b_1}^{b_n} (1 - \alpha_{n,\nu}) \approx \sum_{n=1}^{N} \sum_{\nu=b_1}^{b_n} \alpha_{n,\nu} \tag{15}$$

where $\alpha_{n,\nu}$ is the probability that a DCT difference value $\mathbf{\Delta\tilde{F}_{p,q}}(\nu)$ triggers the false alarm. That is

$$\alpha_{n,\nu} = \begin{cases} P\left[\mathbf{\Delta\tilde{F}_{p,q}}(\nu) - \hat{k} < -\tau\right] \\ \qquad \text{given } \mathbf{\Delta F_{p,q}}(\nu) \geq k, \\ P\left[\mathbf{\Delta\tilde{F}_{p,q}}(\nu) - \hat{k} > \tau\right] \\ \qquad \text{given } \mathbf{\Delta F_{p,q}}(\nu) < k. \end{cases} \tag{16}$$

Because of symmetry, these two probabilities are the same. To calculate $\alpha_{n,\nu}$, we first define a discrete random variable $N'_{d,p}$, s.t., $\lfloor f_p + 1/2 \rfloor + N'_{d,p} \equiv \text{Integer Round}((\mathbf{F_p}(\nu) + N_{d,p})/\mathbf{Q}(\nu)) = \lfloor(\mathbf{F_p}(\nu) + N_{d,p})/\mathbf{Q}(\nu) + 1/2\rfloor$ where $f_p = \mathbf{F_p}(\nu)/\mathbf{Q}(\nu)$ and $\lfloor\cdot\rfloor$ represents the "floor" function. $N'_{d,p}$ is the noise effect in the quantized coefficient, and can be derived from the continuous noise N_d. Its probability density function (pdf) is

$$\begin{aligned} P[N'_{d,p} = n_d] &\\ = P\Bigg[\left(n_d + \left\lfloor f_p + \frac{1}{2}\right\rfloor - f_p + \frac{1}{2}\right) \cdot \mathbf{Q}(\nu) &> N_{d,p} \\ \geq \left(n_d + \left\lfloor f_p + \frac{1}{2}\right\rfloor - f_p - \frac{1}{2}\right) \cdot \mathbf{Q}(\nu)\Bigg]&. \end{aligned} \tag{17}$$

The pdf of $N'_{d,q}$ can be obtained in a similar way. After some transformations from (16)

$$\alpha_{n,\nu} = P\left[N'_{d,p} - N'_{d,q} < \hat{k}' - \tau' - \left\lfloor f_p + \frac{1}{2}\right\rfloor + \left\lfloor f_q + \frac{1}{2}\right\rfloor\right] \tag{18}$$

where $\hat{k}' = \hat{k}/\mathbf{Q}(\nu)$ and $\tau' = \tau/\mathbf{Q}(\nu)$. Then, we can obtain $\alpha_{n,\nu}$ by using the pdf in (17).

Applying (15), the user of image authenticator can set suitable tolerance value τ depending on the quantization table reconstructed from the bitstream, the estimated variances of noise, and the thresholds. In practical applications, the user has to assume models for the pdf of N_r *a priori*; for instance, Gaussian distribution. If the model of N_r is not available, a practical rule is to set τ to *zero* or $\mathbf{Q}(\nu)$. The former is suitable for authenticating one-time compressed images while the latter is better for images that may be recompressed several times.

B. Manipulation and the Probability of Miss

The probability of miss represents the *reliability* of the authenticator. To obtain the probability of miss of a manipulated block pair, we may assume the block p of the image is manipulated and its corresponding block q is untouched. From the viewpoint of the signature generator, any manipulation on the block p of image can be modeled as an additive random variable matrix $\mathbf{M_p}$, *s.t.*

$$\mathbf{\hat{f}_p}(\nu) = \text{Integer Round}\left(\frac{\mathbf{F_p}(\nu) + \mathbf{M_p}(\nu) + N_d}{\mathbf{Q}(\nu)}\right) + N_r \tag{19}$$

where N_d and N_r are computation noises described above. In general, M_p is much larger than N_d and N_r. Therefore, the difference value of the DCT block pair is

$$\begin{aligned} \mathbf{\Delta\hat{F}_{p,q}}(\nu) = \Bigg[&\text{Integer Round}\left(\frac{\mathbf{F_p}(\nu) + \mathbf{M_p}(\nu)}{\mathbf{Q}(\nu)}\right) \\ &- \text{Integer Round}\left(\frac{\mathbf{F_q}(\nu)}{\mathbf{Q}(\nu)}\right)\Bigg] \cdot \mathbf{Q}(\nu). \end{aligned} \tag{20}$$

From Section III-B, we know that the range of $\mathbf{\Delta F_{p,q}}(\nu)$ is bounded by the thresholds used in different sets of the authentication signature. Assume, in the position ν, the range of DCT coefficients is divided into K ranges by the thresholds of the authentication signature. The upper bound and the lower bound of a range are k_l and k_u. For instance, if there is only one threshold, k, then $[k_{l,1}, k_{u,1}) = [-\infty, k)$ in the first range and $[k_{l,2}, k_{u,2}) = [k, \infty)$ in the second range. Assume a coefficient $\mathbf{\Delta F_{p,q}}(\nu)$ is in this range

$$\mathbf{\Delta F_{p,q}}(\nu) \in [k_{l,\nu}, k_{u,\nu}). \tag{21}$$

After JPEG compression, the range of $\mathbf{\Delta\tilde{F}_{p,q}}(\nu)$ should be bounded by $[\hat{k}_{l,\nu}, \hat{k}_{u,\nu}]$ within a tolerance level, τ. Therefore, the probability that the authenticator fails to detect a manipulation on position ν of the block pair (p, q) is

$$\beta_\nu = P\left[\mathbf{\Delta\hat{F}_{p,q}}(\nu) \in \left[\hat{k}_{l,\nu} - \tau, \hat{k}_{u,\nu} + \tau\right]\right]$$

given

$$\mathbf{\Delta F_{p,q}}(\nu) \in [k_{l,\nu}, k_{u,\nu}). \tag{22}$$

If we consider $\mathbf{M_p}$ as a random variable and apply (20), (22) becomes

$$\beta_\nu = P[m_{l,\nu} \leq \mathbf{M_p}(\nu) \leq m_{u,\nu}] \tag{23}$$

where

$$\begin{cases} m_{l,\nu} = \hat{k}_{l,\nu} - \tau + \left(\left\lfloor \dfrac{\mathbf{F_q}(\nu)}{\mathbf{Q}(\nu)} + \dfrac{1}{2}\right\rfloor - \dfrac{1}{2}\right) \\ \qquad \cdot\mathbf{Q}(\nu) - \mathbf{F_p}(\nu), \\ m_{u,\nu} = \hat{k}_{u,\nu} + \tau + \left(\left\lfloor \dfrac{\mathbf{F_q}(\nu)}{\mathbf{Q}(\nu)} + \dfrac{1}{2}\right\rfloor + \dfrac{1}{2}\right) \\ \qquad \cdot\mathbf{Q}(\nu) - \mathbf{F_p}(\nu). \end{cases} \tag{24}$$

Assume a $b_n \times 1$ vector $\mathbf{\hat{M}_p}$, which is a subset of $\mathbf{M_p}$ representing the selected b_n elements of $\mathbf{M_p}$, has a pdf $f(\mathbf{\hat{M}_p})$. Also, assume a range set $\mathbf{RB}$, $\mathbf{RB} = \{\mathbf{\hat{M}_p}(\nu)\colon m_{l,\nu} \leq \mathbf{\hat{M}_p}(\nu) \leq m_{u,\nu}\}, \forall \nu$, to specify the accepted range of manipulation. Then, the probability of miss P_m of a specific image block pair is

$$P_m = \int_{\mathbf{RB}} f\left(\mathbf{\hat{M}_p}\right) d\mathbf{\hat{M}_p}. \tag{25}$$

To derive P_m, we need to know the pdf of manipulation, i.e., $f(\mathbf{\hat{M}_p})$. We first consider manipulations in the spatial domain. Since the possible manipulation to an image block is arbitrary, from the signature generator's viewpoint, there is no exact distribution function. However, we can assume that the manipulated image block will be similar to its adjacent blocks, otherwise this manipulated image block will cause a noticeable artificial effect, which is easily detectable by people. Thus, we may use a multivariate zero-mean Gaussian distribution $\mathbf{\Delta X_p}$: $N[\mathbf{0}, \sigma^2\mathbf{R}]$ to model the probability of additive intensity change of each pixel in the block. The variance parameter σ^2 depends on what kind of manipulation is expected. Some experimental values are shown in Table IV. $\mathbf{R}$ is the covariance matrix of the pixels in a block. In the DCT domain, we can get the probability distribution of $\mathbf{M_p}$ as follows:

$$\mathbf{M_p}\text{: } N[\mathbf{0}, \sigma^2\mathbf{DRD}^t] \tag{26}$$

where $\mathbf{D}$ is the DCT transform matrix defined in (1).

To evaluate an authentication system, we can calculate the probability of miss based on the two extreme cases of $\mathbf{\Delta X_p}$, uncorrelated and fully correlated. In the uncorrelated case, i.e., $\mathbf{R} = \mathbf{I}$, manipulations on each pixels are totally uncorrelated. They are similar to additive white Gaussian noise. Therefore, $\mathbf{M_p}$: $N[\mathbf{0}, \sigma^2\mathbf{I}]$ because $\mathbf{DD}^t = \mathbf{I}$ in DCT. In this case, the probability of miss P_m will be

$$P_m = \prod_{\nu=b_1}^{b_n} \beta_\nu = \prod_{\nu=b_1}^{b_n} \left[\Phi\left(\frac{m_{u,\nu}}{\sigma}\right) - \Phi\left(\frac{m_{l,\nu}}{\sigma}\right)\right] \tag{27}$$

where $\Phi(\cdot)$ is the *standard normal distribution function*, $\Phi(z) = \int_{-\infty}^{z}(1/\sqrt{2\pi})e^{-u^2/2}\,du$. In the fully correlated case, assume there is no weighting on specific positions in the pixel domain. The intensity change of each pixel is the same, i.e., $\mathbf{R} = [r_{ij}\text{: } i, j = 1, \ldots, 64]$ where $r_{ij} = 1$. Then, $\mathbf{M_p}$: $N[\,\mathbf{0}, \sigma^2\hat{\mathbf{R}}]$ where $\hat{\mathbf{R}} = [r_{ij}\text{: } i, j = 1, \ldots, 64]$ with $r_{1,1} = 64$ and $r_{i,j} = 0$, *elsewhere*. In this case, P_m will be

$$P_m = \Phi\left(\frac{m_{u,1}}{8\sigma}\right) - \Phi\left(\frac{m_{l,1}}{8\sigma}\right). \tag{28}$$

Given a specific image block pair with the quantization table, the tolerance values, and the thresholds, we can use (27), (28), and Table IV to estimate the range of probability of miss in an image block pair.

The above computation estimates the probability of miss for a single block pair. In some applications, the authenticator does not need to localize the manipulation. In these cases, the miss probability for the whole image is the product of the miss probabilities of all manipulated block pairs.

C. The Probability of Attack Success

From the attackers' point of view, they want to know the chance of success in attacking the authentication system. There are two kinds of attack. First, attackers may manipulate the image to generate a new image with different visual meaning. In this case, the attacker may use replacement, deletion, or cloning to change the pixel values. This kind of manipulation strategy attempts to blend the replaced pixels smoothly with adjacent areas. Second, attackers may manipulate the image (or synthesize an image) based on their knowledge about the authentication algorithm and secret information in the signature. This strategy is to generate a different image to fool the authenticator. Note the image content may be clearly altered or distorted. In particular, if the attack is done in the DCT domain, noticeable distortion usually can be found in the pixel domain. In the following, we analyze the probabilities of success for these two types of attacks.

1) Attacks with Visual Meaning Changes: Changing visual meaning of an image is a common attack. Based on the changes and estimation of authentication parameters, an attacker can estimate his chance of success. For instance, DCT values of the changed blocks are known to the attacker. If the image will not be further recompressed, the quantization table is also known. Otherwise, he can estimate the range of success probability based on different quantization tables. The threshold values can be estimated by looking at the DCT values and the signature. If this is not available, the first threshold value can be assumed to be zero. The tolerance values used in the authenticator are unknown. But he can assume some reasonable values such as zero or $\mathbf{Q}(\nu)$, and observe their effects on authentication. The only random part for estimating the probability of success would be the values of the DCT coefficients in another block of the pair.

Therefore, the probability of success P_s of a manipulated block can be modeled as

$$P_s = \prod_{\nu=1}^{b_n} \gamma_\nu \tag{29}$$

where γ_ν is the probability of success for each DCT coefficient. We can compute γ_ν as follows:

$$\begin{aligned}
\gamma_\nu =& \sum_K \Big\{P\Big[\mathbf{\Delta\hat{F}_{p,q}}(\nu) - \hat{k}_l \geq -\tau, \mathbf{\Delta F_{p,q}}(\nu) \geq k_l\Big] \\
&\quad + P\Big[\mathbf{\Delta\hat{F}_{p,q}}(\nu) - \hat{k}_u \leq \tau, \mathbf{\Delta F_{p,q}}(\nu) < k_u\Big]\Big\} \\
=& \sum_K \Big\{P\Big[\mathbf{F_q}(\nu) \leq \left(\mathbf{\hat{f}_m}(\nu) + \frac{1}{2}\right)\mathbf{Q}(\nu) - \hat{k}_l + \tau, \\
&\qquad \mathbf{F_q}(\nu) \leq \mathbf{F_p}(\nu) - k_l\Big] \\
&\quad + P\Big[\mathbf{F_q}(\nu) \geq \left(\mathbf{\hat{f}_m}(\nu) - \frac{1}{2}\right)\mathbf{Q}(\nu) - \hat{k}_u - \tau, \\
&\qquad \mathbf{F_q}(\nu) > \mathbf{F_p}(\nu) - k_u\Big]\Big\} \\
\equiv& \sum_K \gamma_{K,\nu}
\end{aligned} \tag{30}$$

where

$$\mathbf{\hat{f}_m}(\nu) = \left\lfloor \frac{\mathbf{F_p}(\nu) + \mathbf{M_p}(\nu)}{\mathbf{Q}(\nu)} + \frac{1}{2} \right\rfloor. \tag{31}$$

To estimate P_s, the attacker can assume $\mathbf{F_q}(\nu)$ to be a random variable with a Gaussian distribution with a zero mean and a variance of σ_ν^2. Therefore, $P[\mathbf{F_q}(\nu) \leq \mathbf{F_p}(\nu) - k_l]$ can be written as $\Phi((\mathbf{F}_{\mathbf{p}}(\nu) - k_l)/\sigma_\nu)$. Other probabilities can be approximated in a similar way. We can obtain the success probability of each coefficient, $\gamma_{\kappa,\,\nu}$, as

$$\gamma_{\kappa,\,\nu} = \begin{cases} \min\left[0,\ \Phi\left(\dfrac{\left(\hat{\mathbf{f}}_{\mathbf{m}}(\nu) + \frac{1}{2}\right)\mathbf{Q}(\nu) - \hat{k}_l + \tau}{\sigma_\nu}\right) \right. \\ \qquad \left. - \Phi\left(\dfrac{\mathbf{F}_{\mathbf{p}}(\nu) - k_u}{\sigma_\nu}\right)\right], \\ \qquad\qquad \hat{\mathbf{f}}_{\mathbf{m}}(\nu) < \dfrac{\mathbf{F}_{\mathbf{p}}(\nu) - \tau + \hat{k}_l - k_l}{\mathbf{Q}(\nu)} - \dfrac{1}{2} \\ \Phi\left(\dfrac{\mathbf{F}_{\mathbf{p}}(\nu) - k_l}{\sigma_\nu}\right) - \Phi\left(\dfrac{\mathbf{F}_{\mathbf{p}}(\nu) - k_u}{\sigma_\nu}\right), \\ \qquad\qquad \text{elsewhere}, \\ \min\left[0,\ \Phi\left(\dfrac{\mathbf{F}_{\mathbf{p}}(\nu) - k_l}{\sigma_\nu}\right) \right. \\ \qquad \left. - \Phi\left(\dfrac{\left(\hat{\mathbf{f}}_{\mathbf{m}}(\nu) - \frac{1}{2}\right)\mathbf{Q}(\nu) - \hat{k}_u - \tau}{\sigma_\nu}\right)\right], \\ \qquad\qquad \hat{\mathbf{f}}_{\mathbf{m}}(\nu) > \dfrac{\mathbf{F}_{\mathbf{p}}(\nu) + \tau + \hat{k}_u - k_u}{\mathbf{Q}(\nu)} + \dfrac{1}{2}. \end{cases} \tag{32}$$

It should be noticed that the attacker has to calculate the DCT values of the manipulated blocks and estimate σ_ν^2 before applying (29)–(32).

From (32), we can observe that $\forall\,\nu$, if $\hat{\mathbf{f}}_{\mathbf{m}}(\nu) \in [(\mathbf{F}_{\mathbf{p}}(\nu) - \tau + \hat{k}_l - k_l)/\mathbf{Q}(\nu) - 1/2,\ (\mathbf{F}_{\mathbf{p}}(\nu) + \tau + \hat{k}_u - k_u)/\mathbf{Q}(\nu) + 1/2]$ in all ranges, then the probability of success P_s will be equal to 1. Using transformations similar to those in (23), we can represent this range in the DCT domain

$$\begin{aligned} &\left\{\left\lfloor \frac{\mathbf{F}_{\mathbf{p}}(\nu) - k_l}{\mathbf{Q}(\nu)} + \frac{1}{2}\right\rfloor - \frac{1}{2} - \frac{\mathbf{F}_{\mathbf{p}}(\nu) - \hat{k}_l}{\mathbf{Q}(\nu)}\right\} \cdot \mathbf{Q}(\nu) - \tau \\ &\quad \le \mathbf{M}_{\mathbf{p}}(\nu) \\ &\quad < \left\{\left\lfloor \frac{\mathbf{F}_{\mathbf{p}}(\nu) - k_u}{\mathbf{Q}(\nu)} + \frac{1}{2}\right\rfloor + \frac{1}{2} - \frac{\mathbf{F}_{\mathbf{p}}(\nu) - \hat{k}_u}{\mathbf{Q}(\nu)}\right\} \cdot \mathbf{Q}(\nu) + \tau. \end{aligned} \tag{33}$$

Equation (33) specifies the range in which an attacker can change the coefficients without triggering the authentication alarm. Note the size of this undetected manipulation range is equal to $\mathbf{Q}(\nu)$.

We can rewrite the above range as

$$\mathbf{M}_{\mathbf{p}}(\nu) \in [a - \mathbf{Q}(\nu),\ a] \tag{34}$$

where a is a coefficient dependent variable within the range of $[\tau - 1.5\mathbf{Q}(\nu),\ \tau + 2.5\mathbf{Q}(\nu)]$. Given that τ and $\mathbf{Q}(\nu)$ are unknown, the attacker cannot determine a fixed bound for undetected manipulations. Therefore, an attacker has no way to maliciously manipulate an image without taking the risk of triggering the authentication alarm.

2) Attacks with Knowledge of Authentication Rules: Some attackers may try to manipulate an image based on their knowledge about the authentication techniques, but regardless of the visual meaning of the manipulated image. Attackers may want to manipulate or even synthesize an image that can fool the system without triggering the alarm. In our authentication system, the security mechanism is based on: 1) the private key used for the signature encryption, which ensures the signature cannot be forged; 2) the secret transformation mechanism and a seed to generate the mapping function for selecting the block pairs; and 3) the secret method and another seed used to select DCT coefficient positions in block pairs for comparison. In the following paragraphs, we will discuss four possible situations, with different extent of knowledge possessed by the attacker.

Security Level I: All Information in the Signature is Secret: If all information in the signature is kept secret from the attacker, the performance of the proposed authenticator is the highest, as analyzed in previous sections. The only possible attack is to make a constant change to DCT coefficients at the same location in all blocks. We have proposed a way to solve this problem by recording the mean values of DCT coefficients as discussed in Section III-B-1 and Section III-C.

Security Level II: The Selected DCT Coefficient Positions are Known: The locations of the selected block pairs and the DCT coefficients are determined by some secret algorithms, which are in turn driven by random seed numbers. The secret algorithms are usually pre-designed by the manufacturer of the authentication system. They can be stored as secret bytecodes embedded in the system. Therefore, even though the random seeds can be known by the attacker, the real selected positions are still unknown to the attacker.

In a pessimistic scenario, the attacker knows the secret algorithms and seeds for the selected DCT coefficients. Once the knows the real selected positions, he can arbitrarily change the coefficients that are not compared in the authentication process without triggering the authentication alarm. To avoid this problem, the authenticator can change the rule of selected positions, block by block, in a more complicated method. Furthermore, if this threat persists, the signature generator can eventually use all the 64 DCT coefficients in each block.

Security Level III: The Mapping Function of Block Pairs is Known: Once the mapping function is known, the attacker also knows the DCT differences for each pair of blocks. For example, if only the sign of the DCT differences are used for authentication, and the attacker knows $\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) = 10$ in the original compressed image, he can manipulate this value to $\mathbf{\Delta}\hat{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) = 60$, which will not be detected by the authenticator. In this case, multiple threshold sets $\mathbf{k}$ should be used because they can protect each coefficient with a higher accuracy. Although the DCT differences are known to the attacker, he still cannot manipulate those DCT coefficients too much, because the allowed degree of manipulation is reduced as more bits i.e., smaller k values) are used.

Security Level IV: The Private Key used for Signature Encryption is Known: The use of the private key ensures that only

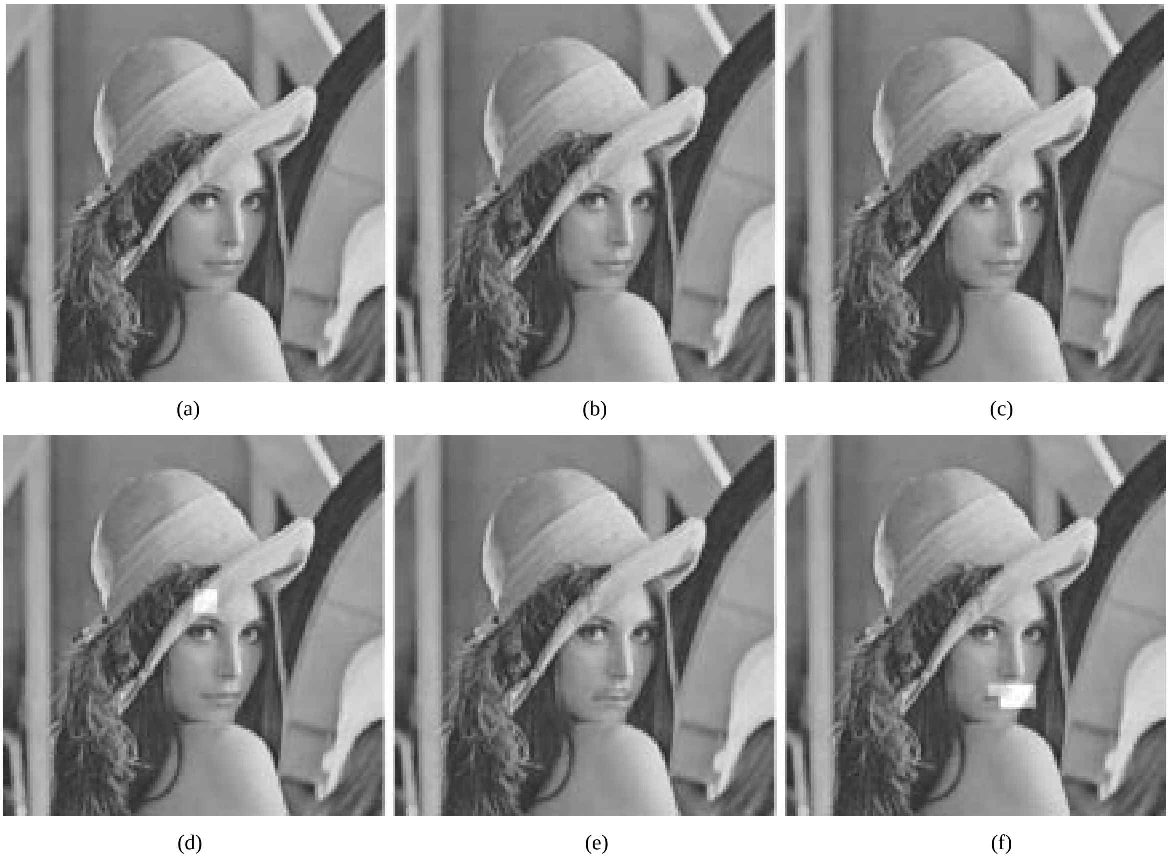


Fig. 4. (a) Original image. (b) JPEG compressed image (compression ratio 9 : 1). (c) Middle of hat brim cloned. (d) Authentication result of (c). (e) Mouth manipulated. (f) Authentication result of (e).

the right source can generate the authentication signature. In the extreme hypothetical case, the private key used by the original source may be known to the attacker. This is a general problem for any secure communication and is out of the scope of this paper. However, one possible way to solve this problem is to ask the original source to register and store its signature in a trustable institution. The institution stamps a digital postmark on the signature to prove its receiving time and its originality. Therefore, the forged signature will be considered invalid because its originality cannot be proven.

It is also worth noting that subjective inspection may provide another means of protecting the image authenticity. The attacker may try to develop special manipulations in the DCT domain in order to break the proposed scheme. But, at the same time, it is difficult for the attacker to control the resulting artifacts in the pixel domain. These artifacts may be very obvious to humans, even as they are able to circumvent the authentication process.

V. Experimental Results

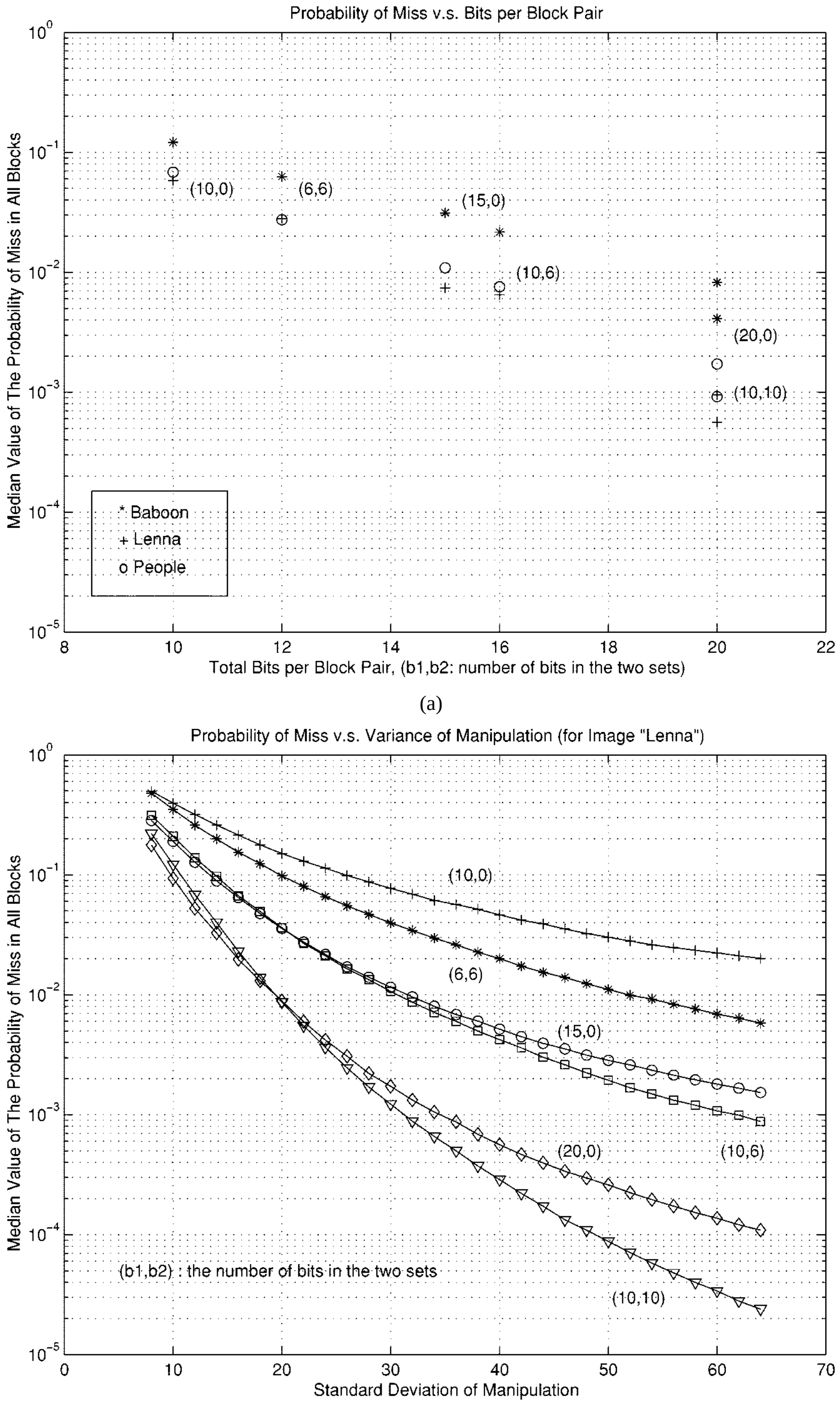
A. Experiments

In evaluating the proposed image authenticator, we test different manipulations on the well-known “Lenna” image. The original image is shown in Fig. 4(a). In our experiment, the authentication results together with the DCT coefficients $\hat{F}$ are sent to an IDCT to convert those coefficients to the pixel domain. Those blocks detected as manipulated will be highlighted, with the highlight intensity proportional to the number of manipulated coefficients in that block. Therefore, the more coefficients modified, the brighter this block will be. There are 10 bits per block pair used in generating the signature codes.

Experiment 1: Lossy Compression: The “Lenna” image is compressed with compression a ratio of 9 : 1. The authentication signature is generated based on the original “Lenna” image. The compressed bitstream is sent to the system for authentication. The tolerance bound of the authenticator is set to $\tau = 0$, since no integral rounding is involved. As previously predicted, the authenticator will verify the compressed image as authentic and decompress this image perfectly. The authentication result is shown in Fig. 4(b).

Experiment 2: Recompression and Integer Rounding: The original image is compressed with a compression ratio 6 : 1. Then, this image is decompressed by Photoshop, rounded to integral values, and recompressed into an image with compression ratio of 9 : 1. In this case, if we use $\tau = \mathbf{Q}(\nu)$, the recompression process (9 : 1) will not trigger the manipulation detector and the final compressed image is still verified as authentic. The final decoded image is similar to Fig. 4(b).

Experiment 3: Detection of Manipulation: The third experiment is made by manipulating the image by deleting the feather fringe hanging over the hat brim, just above Lenna’s right eye. This feather area (16×16 pixels) is removed and cloned by its neighboring pixels. This image is shown in Fig. 4(c). The authentication result is shown in Fig. 4(d). It is clearly shown that the manipulated part has been detected as fake; it is highlighted



(a)

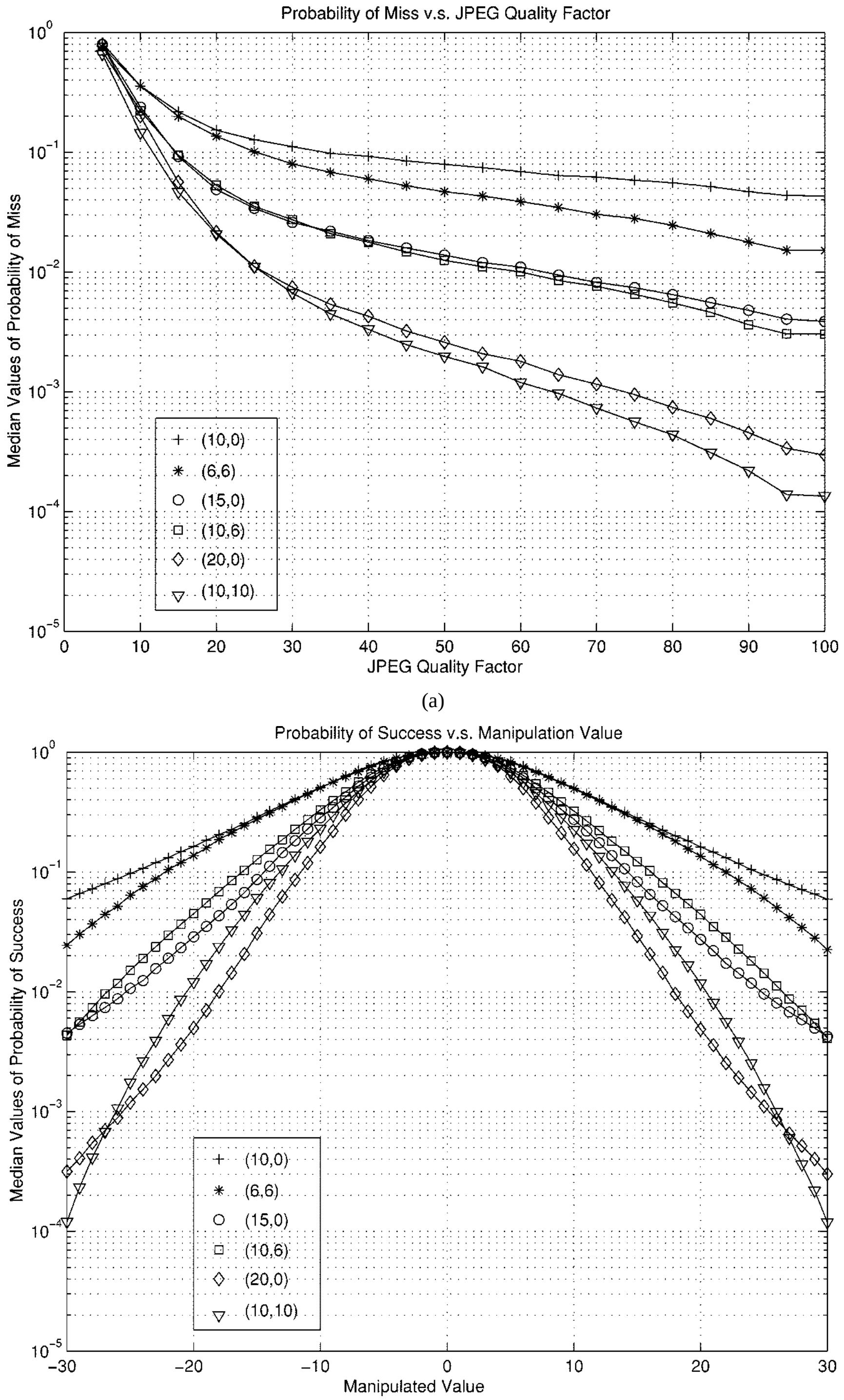
(b)

Fig. 5. Probability of: (a) miss with different images and (b) miss with different signature lengths.

by the authenticator. The other example is shown in Fig. 4(e). In this image, Lenna's mouth was flipped in the vertical direction. Its authentication result is shown in Fig. 4(f).

B. Probability of Miss and Attack Success

From Figs. 5 and 6, we evaluate practical system performance by analyzing the probability of miss and the probability of success in different cases. Fig. 5(a) shows the median values of the probability of miss in several images. The tolerance value $\tau = 0$, the threshold values $k = 0, \pm 16$, and the standard deviation of manipulations (35) are used in this figure. (If not specified, these settings are kept the same for other figures.) In these figures, a (b_1, b_2) symbol means b_1 bits are used in the first set of the feature codes, and b_2 are used in the second set. For instance, 10 bits used per block pair are denoted by a $(10, 0)$ symbol.



(a)

(b)

Fig. 6. (a) Probability of miss of images with different JPEG quality factors. (b) Probability of duccess with different manipulation values.

TABLE IV
STANDARD DEVIATION OF DIFFERENT OPERATIONS (RESULTS OF EXPERIMENTS USING PHOTOSHOP 3.0 TO MANIPULATE IMAGE IN THE PIXEL DOMAIN)

Image	Replacement	Blur	Sharpen	Histogram Equalization
Lenna	25.8 – 55.0	8.7 – 10.7	9.43 – 12.7	23.1

Fig. 5(b) shows the probability of miss with different standard deviations of manipulations. These values are derived from (27) and (28). Referring to Table IV, the standard deviation of a replacement manipulation is between 25.8 and 55.0. Through our experiments of ten images, the medain value of this change is between 35 and 40. If we use 40 as the possible standard deviation of malicious manipulation, the estimated value of P_m will be 0.04 for a $(10, 0)$ signature or 0.0004 for a $(20, 0)$ signature. The JPEG quality factor is 75, in this case.

Although the same authentication system is valid regardless of the image compression rate, the probability of miss is variable because the allowable modification range is increased as the quality factor decreases. This is shown in Fig. 6(a). We use (27) and (28) to compute these values. The standard deviation of manipulation is set to 35. We can see that P_m increases when the image is compressed more.

Because the probability of success is case dependent, it can only be estimated when the attacker's actual manipulations are given. It is impractical to compute a single universal P_s for an image. However, as an example, we change all DCT coefficients in a block with a constant and compute the P_s. Then we vary to location of the changed block in the image. For each block, we obtain a P_s. Finally, the median value of all probabilities versus the change magnitudes is shown in Fig. 6(b). For instance, if we use the $(15, 0)$ signature and increase each DCT coefficient in a block by 20, then the probability of sucess is about 0.03. In other words, assuming the attacker knows some authentication parameters $(\mathbf{Q}(\nu), \tau)$ but does not know which blocks are formed as pairs, his manipulation attack has a 0.03 probability of success.

Observing these figures, we know that the more bits used, the less the probability of miss will be. Also, we know that if the same number of bits was used, the performance of authentication signatures with two threshold sets will be better than those with only one set.

VI. Conclusion

In this paper, we have proposed an image authentication technique that distinguishes the JPEG lossy baseline compression from other malicious manipulations. In practical applications, images may be compressed and decompressed several times and still considered as authentic. Some manipulations, e.g., integral value rounding, color space transformation and cropping, are also considered acceptable in some applications. We propose a technique that allows JPEG lossy compression but prevents malicious manipulations. Our proposed technique can be customized to accommodate different requirements and accept "desirable" manipulations. Our extensive analytic and empirical performance analysis has shown the effectiveness of this system.

Appendix I
Proof of Theorem 1 and Theorem 2

Proof 1: $\forall\, a, b, c \in \Re$, assume $a = A + r(a)$, $b = B + r(b)$, and $c = C + r(c)$, where $A, B, C \in Z$ are the rounding integers of a, b, c, respectively, and $-0.5 \le r(a), r(b), r(c) < 0.5$. Assume $a - b > c$, then

$$A + r(a) - B - r(b) > C + r(c). \tag{35}$$

Therefore

$$A - B - C > r(c) + r(b) - r(a). \tag{36}$$

If c is an integer, i.e., $r(c) = 0$, then

$$A - B - C > -1.0. \tag{37}$$

Since A, B, C are integers

$$A - B \ge C. \tag{38}$$

If $r(c) \ne 0$, then $-1.5 < r(c) + r(b) - r(a) < 1.5$. Since $A, B, C \in Z$

$$A - B \ge C - 1. \tag{39}$$

Theorem 1 can be proved by substituting a by $\mathbf{F}_\mathbf{p}(u, v)/\mathbf{Q}(u, v)$, A by $\tilde{\mathbf{F}}_\mathbf{p}(u, v)/\mathbf{Q}(u, v)$, b by $\mathbf{F}_\mathbf{q}(u, v)/\mathbf{Q}(u, v)$, B by $\tilde{\mathbf{F}}_\mathbf{q}(u, v)/\mathbf{Q}(u, v)$, c by 0, and with every parameter multiplied by $\mathbf{Q}(u, v)$. In Theorem 2, (5) can be proved by the same parameter substitutions except c is replaced by $k/\mathbf{Q}(u, v)$ and C is replaced by $\tilde{k}_{u,v}$. Equations (6) and (7) can be proved by using similar methods. □

In some software implementations, the integer rounding process is replaced by the truncation process. In this case, Theorem 1 and 2 are still valid. They can be proved by *Proof 2* with the same parameter substitutions as in *Proof 1*.

Proof 2: $\forall\, a, b, c \in \Re$, assume $a = A + r(a)$, $b = B + r(b)$, and $c = C + r(c)$, where $A, B, C \in Z$ are the truncated integers of a,b,c, respectively, and $0 \le r(a), r(b), r(c) < 1$. Similarly, in the case that $a - b > c$, i.e., $A - B - C > r(c) + r(b) - r(a)$, if c is an integer, then $-1.0 < r(c) + r(b) - r(a) < 1.0$. Therefore

$$A - B \ge C. \tag{40}$$

If $r(c) \ne 0$, then $-1 < r(c) + r(b) - r(a) < 2$. Since $A, B, C \in Z$

$$A - B - C \ge 0 > -1 \tag{41}$$

and therefore

$$A - B \ge C \tag{42}$$

which satisfies $A - B \ge C - 1$. □

Appendix II
Variable Quantization Tables

In some image/video compression techniques, different quantization tables are used in different image blocks for adaptive compression rate control, such as in MPEG or later JPEG standards. In these cases, the proposed image authentication techniques can be extended by the following theorem.

Theorem 3: Use the parameters defined in Theorem 1, except $\tilde{\mathbf{F}}_\mathbf{p}$ is defined as $\tilde{\mathbf{F}}_\mathbf{p}(\nu) \equiv \text{Integer Round}(\mathbf{F}_\mathbf{p}(\nu)/\mathbf{Q}_\mathbf{p}(\nu)) \cdot \mathbf{Q}_\mathbf{p}(\nu)$ and $\tilde{\mathbf{F}}_\mathbf{p}(\nu) \equiv \text{Integer Round}(\mathbf{F}_\mathbf{q}(\nu)/\mathbf{Q}_\mathbf{q}(\nu)) \cdot \mathbf{Q}_\mathbf{q}(\nu)$, where $\mathbf{Q}_\mathbf{p}$ and $\mathbf{Q}_\mathbf{q}$ are quantization tables for blocks $\mathbf{F}_\mathbf{p}$ and $\mathbf{F}_\mathbf{q}$ respectively. Assume a fixed threshold $k \in \Re$. The following properties hold:

1) if $\mathbf{\Delta F}_{\mathbf{p},\mathbf{q}}(\nu) \ge k$, then $\mathbf{\Delta}\tilde{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) \ge k - 1/2 \cdot (\mathbf{Q}_\mathbf{p}(\nu) + \mathbf{Q}_\mathbf{q}(\nu))$;
2) else if $\mathbf{\Delta F}_{\mathbf{p},\mathbf{q}}(\nu) < k$, then $\mathbf{\Delta}\tilde{\mathbf{F}}_{\mathbf{p},\mathbf{q}}(\nu) \le k + 1/2 \cdot (\mathbf{Q}_\mathbf{p}(\nu) + \mathbf{Q}_\mathbf{q}(\nu))$. □.

We redefine (11) as

$$\hat{k} = \begin{cases} k + \dfrac{1}{2}(\mathbf{Q_p}(\nu) + \mathbf{Q_q}(\nu)), & \text{if } Z_n(\nu) = 0, \\ & \text{i.e., } \mathbf{\Delta F_{p,q}}(\nu) < k \\ k - \dfrac{1}{2}(\mathbf{Q_p}(\nu) + \mathbf{Q_q}(\nu)), & \text{if } Z_n(\nu) = 1, \\ & \text{i.e., } \mathbf{\Delta F_{p,q}}(\nu) \geq k. \end{cases}$$

In other words, if $\mathbf{\Delta F_{p,q}}(\nu) < k$, then $\mathbf{\Delta \hat{F}_{p,q}}(\nu) - \hat{k} \leq 0$ must be satisfied.

Except for the above modifications, the authentication system designed for the variable quantization table cases would be the same as the proposed system for the case with equal quantization tables. A detailed discussion of this case is in [14].

References

[1] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998.

[2] G. W. Braudaway, K. A. Magerlein, and F. Mintzer, "Protecting publicly-available images with a visible image watermark," IBM T.J. Watson Research Center, Yorktown Heights, NY, Res. Rep. RC 20336, Jan. 1996.

[3] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," NEC Research Institute, Princeton, NJ, Tech. Rep. 95-10, 1995.

[4] S. Craver, N. Memon, B. L. Yeo, and M. Yeung, "Can invisible watermarks resolve rightful ownerships," IBM T.J. Watson Research Center, Yorktown Heights, NY, Res. Rep. RC 20509, July 1996.

[5] W. Diffle and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[6] J. Fridrich, "Image watermarking for tamper detection," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998.

[7] ——, "Methods for detecting changes in digital images," in *Proc. IEEE Workshop on Intelligent Signal Processing and Communication Systems*, Melbourne, Australia, Nov. 1998.

[8] G. L. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic image," *IEEE Trans. Consumer Electron.*, vol. 39, pp. 905–910, Nov. 1993.

[9] IETF Networking Group. (1999, Mar.) *Internet X.509 public key infrastructure certificate management protocols*. Internet Engineering Task Force (IETF), RFC 2510. [Online]. Available: http://www.ietf.org /html.charters/pkix-charter.html

[10] (1998, August) *Public key infrastructure: The PKIX reference implementation project*. IBM Secureway White Paper. [Online]. Available: http://www.ibm.com/security/html/wp-pkix.pdf

[11] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989, pp. 80–99.

[12] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*. Englewood Cliffs, NJ: Prentice-Hall, 1995, pp. 129–162.

[13] C.-Y. Lin and S.-F. Chang, "A robust image authentication method surviving JPEG lossy compression," *SPIE Storage and Retrieval of Image/Video Databases*, Jan. 1998.

[14] ——, "An image authenticator surviving DCT-based variable quantization table compressions," CU/CTR, New York, Tech. Rep. 490-98-24, Nov. 1997.

[15] ——, "A watermark-based robust image authentication method using wavelets," Columbia Univ., New York, ADVENT Project Rep., Apr. 1998.

[16] ——, "Issues and solutions for authenticating MPEG video," in *SPIE Conf. Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 1999, pp. 54–65.

[17] C.-Y. Lin. Bibliography of multimedia authentication research papers. [Online]. Available: http://www.ctr.columbia.edu/~cylin/auth/bibauth.html

[18] B. M. Macq and J. J. Quisquater, "Cryptology for digital TV broadcasting," *Proc. IEEE*, vol. 83, pp. 944–957, June 1995.

[19] K. Matsui and K. Tanaka, "Video-steganography: How to secretly embed a signature in a picture," in *Proc. IMA Intellectual Property Project*, vol. 1, 1994, pp. 187–206.

[20] M. P. Queluz, "Content-based integrity protection of digital images," in *SPIE Conf. Security and Watermarking of Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 1999, pp. 85–93.

[21] L. L. Scharf, *Statistical Signal Processing—Detection, Estimation, and Time Series Analysis*. Reading, MA: Addison-Wesley, 1991, pp. 103–178.

[22] M. Schneider and S.-F. Chang, "A robust content based digital signature for image authentication," in *Proc. IEEE Int. Conf. Image Processing*, Laussane, Switzerland, Oct. 1996.

[23] B. Schneier, *Applied Cryptography*. New York: Wiley, 1996, pp. 461–502.

[24] S. D. Silvey, *Statistical Inference*. London, U.K.: Chapman & Hall, 1975, pp. 94–107.

[25] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30–44, Apr. 1991.

[26] S. Walton, "Image authentication for a slippery new age," *Dr. Dobb's J.*, pp. 18–26, April 1995.

[27] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, Oct.r 1997.

[28] J. Zhao and E. Koch, "Embedding robust label into images for copyright protection," in *Proc. Int. Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Vienna, Austria, Aug. 1995.

Ching-Yung Lin (S'98–M'01) received the B.S. and M.S. degrees from National Taiwan University, Taiwan, R.O.C., in 1991 and 1993, respectively, and the Ph.D. degree from Columbia University, New York, in 2000, all in electrical engineering.

During 1993–1995, he was a Wireless Communication Engineer in the Taiwan Air Force. In 2000, he joined the IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member. His current research interests include multimedia authentication and watermarking techniques for security, multimedia indexing and query, multimedia transmission and networking, human–computer interaction, and multirate multidimensional signal processing. He was the primary contributor in the design of the first successful self-authentication-and-recovery multimedia system, which distinguishes JPEG/MPEG compression from malicious manipulation, and in the design of the first public/blind watermarking method surviving print-and-scan process. He holds three pending U.S. patents.

Dr. Lin was the recipient of Lung-Teng Thesis Award and an Outstanding Paper Award, both in 1993.

Shih-Fu Chang (S'89–M'93) received the Ph.D. degree in Electrical Engineering and Computer Science from the University of California at Berkeley in 1993.

He is an Associate Professor of Electrical Engineering at Columbia University, and currently leads Columbia's ADVENT Industry–University Research Consortium, which focuses on representation, manipulation, searching, and transmission of multimedia content. He also leads digital video research within several cross-disciplinary projects at Columbia, including Columbia's Health Care Digital Library Project supported by the National Science Foundation's DLI Phase-II initiative. He actively participates in international conferences and standardization efforts, such as MPEG-7, and has been Consultant for several new media companies.

Prof. Chang was a General Co-Chair of ACM Multimedia Conference 2000 and an Associate Editor for several journals (AU: PLS. NAME). He was awarded a Faculty Development Award from IBM in 1995, a CAREER Award from the National Science Foundation in 1995, a Navy ONR Young Investigator Award in 1998, and three Best Paper Awards in the areas of video representation and searching.