A Robust and Secure Media Signature Scheme for JPEG Images

(Special Issue for MMSP2002, Guest Editor: Prof. Yun Q. Shi,)

Qibin Sun[1] and Shih-Fu Chang[2]

[1]Institute for Infocomm Research (I^{2}R)

21 Heng Mui Keng Terrace, Singapore 119613

[2]Department of Electrical Engineering, Columbia University

New York City, NY10027, USA

Email: qibin@i2r.a-star.edu.sg / sfchang@ee.columbia.edu

Correspondence address

Qibin Sun

Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613

Email: qibin@i2r.a-star.edu.sg

Phone: +65 - 68746696

Fax: +65 - 67744998

A Robust and Secure Media Signature Scheme for JPEG Images

Qibin Sun and Shih-Fu Chang

Abstract— In [1, 2, 3], we have briefly introduced a robust and secure digital signature solution for multimedia content authentication, by integrating content feature extraction, error correction coding (ECC), watermarking and cryptographic signature into one unified framework. In this paper, we firstly study the difference between two types of distortions, namely incidental distortions which come from the defined acceptable content manipulations such as lossy compression and intentional distortions which come from real attacks such as content copy-paste. We then introduce a new JPEG-compliant solution under our proposed framework but with different ECC and watermarking methods. Finally system performance evaluation and analysis will be given to further demonstrate the practicability of our method.

Index Terms—Image Authentication, Digital Signature, Watermarking, Content hash, ECC

I. INTRODUCTION

Our objective in this paper is to design a cryptographic signature scheme (i.e., digital signature) that allows two parties to exchange images while guaranteeing content integrity and non-repudiation from content owner, in a semi-fragile way. Integrity protection means that the content is not allowed to be modified in a way such that the content meaning is altered. Non-repudiation prevention means that once a content owner generates the content signature, he cannot subsequently deny such a signing if both the signature and the content have been verified as being authentic.

State-of-the-art work shows that the above objective can be achieved in a fragile way (i.e., even one bit change is not allowable) either by watermarking [4, 5] or by cryptographic digital signature scheme such as RSA or DSA [6, 7, 8]. However, some applications demand the same security solution on a semi-fragile level, i.e., some manipulations on the content will be considered acceptable (e.g. lossy compression) while some are not allowable (e.g. content copy-paste). However, at the semi-fragile level, watermarking-based approaches only work well in protecting the integrity of the content [9, 10, 11] but are unable to solve the non-repudiation issue as a symmetric key is used for both watermark embedding and extracting. Once the key or watermark is compromised, attackers can use the key or watermark to fake other images as authentic. Signature based approaches can work on both the integrity protection of the content and the repudiation prevention of the owner, but a shortcoming exists. The generated signature is unavoidably very large because its size is usually proportional to the image size. Some recent work can be found in [12, 13, 14, 15]. More detailed survey on prior work will be presented in Section II.

In [1, 2, 3], we have already introduced a robust and secure digital signature solution for multimedia content-based authentication, by integrating content-based feature extraction, error correction coding, watermarking and crypto signature into one unified framework. The proposed scheme is *efficient* by generating only one crypto signature (hundreds of bits [16]) per image regardless of image size, in a semi-fragile way. System *robustness* (i.e., the ability to tolerate incidental distortions introduced from the predefined acceptable manipulations such as lossy compression) is achieved based on error correction coding (ECC) concepts. System *security* (i.e., the ability to prevent attacked images from passing authentication) is obtained by adopting crypto hashing and signature scheme. In addition, watermarking is used for storing ECC check information and locating attacks.

The above framework is compliant with traditional digital signature system structures. In the image signing procedure, the image owner uses his private key to sign on the hash value of the extracted features to obtain the signature, and embed the watermark to the image. He then sends the watermarked image to the recipients. In the image verification procedure, the recipient can verify the received image's authenticity by using its owner's public key and the associated signature. The watermarking is done in such a way that it can indicate the locations of attacks on the image if the authentication procedure fails. Such a capability is important because it helps to visually convince users of the authentication result. In order to differentiate our framework from traditional cryptographic signature schemes such as RSA, we name our proposed solution as **media signature** hereafter.

The paper is organized as follows. A detailed survey on signature based media authentication is presented in Section II. Section III discusses the incidental distortion from the acceptable manipulations and the intention distortion from attacks. Considering compliance with the JPEG standard encoding and decoding procedure, in Section IV we shall introduce a JPEG-compliant solution derived from our previous solutions [1, 2, 3] but with different ECC and watermarking techniques. System security and robustness will be analyzed and evaluated in Section V. Conclusion and future work are presented in Section VI.

II. PRIOR WORK ON SIGNATURE BASED MEDIA AUTHENTICATION

Figure 1 (a) shows the brief diagram of cryptographic digital signature [6]. Given a message of arbitrary length, a short fixed-length digest is obtained by a crypto hash operation (e.g., 128 bits by MD5 or 160 bits by SHA-1). The signature is generated using the sender's private key to sign on the hashed digest. The original message associated with its signature is then sent to the intended recipients. Later on, the recipient can verify whether a) the received message was altered, and b) the message were really sent from the sender, by using the sender's public key to authenticate the validity of the attached signature. Based on security consideration, the requirements for a crypto hash functions are [4]: a). Given a message m and a hash function H, it should be easy and fast to compute the hash $h = H(m)$. b). Given h, it is hard to compute m such that $h = H(m)$ (i.e., the hash function should be one-way). c). Given m, it is hard to find another data m' such that $H(m') = H(m)$ (i.e., collision free). The final authentication result is then drawn from a bit-bit comparison between two hash codes (one is decrypted from the signature and the other is obtained by re-hashing the received message, refer to Figure 1(a)) by the criterion: Even if for one bit difference the received message will be deemed unauthentic. Such fragile authentication solutions which directly apply crypto signature techniques to image data, could be found in [7, 8].

Originating from the ideas of "fragile" digital signature as described above, Chang, Lin and other researchers [12, 13, 14, 15] proposed to use some typical content-based measures as the selected features for generating content signature, by assuming that those features are insensitive to incidental distortions but sensitive to intentional distortions. Those features include histogram map, edge/corner map, moments and more.

Considering that applying a lossy compression such as JPEG should be deemed an acceptable manipulation in most applications, Lin and Chang [13] discovered a mathematical invariant relationship between two coefficients in a block pair before and after JPEG compression and use it as the selected feature. Similarly, Lu and Liao [14] presented a structural signature solution for image authentication by identifying the stable relationship between a parent-child pair of coefficients in the wavelet domain. These feature based crypto signature schemes are illustrated in Figure 1(b). We can see that the module of "crypto hash" in Figure 1(a) has been replaced with the module of "feature extraction" in order to tolerate some incidental distortions. The replacement is applied because the acceptable manipulations will cause changes to the content features, though the changes may be small compared to content-altering attacks. Such "allowable" changes to the content features make the features non-crypto-hashing. (Any minor changes to the features may cause a significant difference in the hashed code due to the nature of the crypto hashing methods such as MD5 and SHA-1). Accordingly, as a result of the incompatibility with crypto hashing, the generated signature size is proportional to the size of the content, which is usually very large. Because in typical digital signature algorithms the signature signing is more computational than signature verifying, this will also result in a time-consuming signing process because the size of the formed signature is much greater than 320 bits [16] and it has to be broken into small pieces (less than 320 bits) for signing. On the other hand, no crypto hashing on selected features will make the decision of authenticity usually based on comparison of the feature distance (between the original one decrypted from the signature and the one extracted from the received image) against a threshold value, which is hard to determine in practice and brings some potential security risks.

Since directly applying crypto hashing to images (features) seems infeasible and feature-based correlation approaches still do not resolve the issues such as signature size and security risks, other researchers have already been working on designing their own hash functions named **robust hash** or **content hash**, see Figure 1(c). Differing from crypto hash functions, content hashing aims to generate two hash codes with short hamming distance if one image is a corrupted version of another image by incidental distortions. In other words, if two images or two image blocks are visually similar, their corresponding hash codes should be close in terms of hamming distance. For example, in Fridrich and Goljan's [17] solution, the hash function is designed to return 50 bits for each 64x64 image block by projecting this 64x64 image block onto a set of 50 orthogonal random patterns with the same size (i.e., 64x64) generated by a secret key. The final hash value of the image is then obtained by concatenating the hash codes from all 64x64 image blocks.

Xie, Arce and Graveman [18] proposed a content hash solution for image authentication named Approximate Message Authentication Codes (AMAC). The AMAC is actually a probabilistic checksum calculated by applying a series of random XOR operations followed by two rounds of majority voting to a given message. The similarity between two messages can be measured using the hamming distance of their AMACs. The length of an AMAC is typically around 80-400 bits. Venkatesan, Koon, Jakubowski and Moulin [19] also proposed a solution for generating content hash for image. Firstly, the image is randomly tiled and wavelet transform is applied to each tile independently. Some statistics measures such as mean and variance are calculated in each wavelet domain. Secondly those obtained measures are quantized using a random quantizer (i.e., the quantization step size is random) to increase security against attacks. Thirdly, the quantized statistics measures in each tile are decoded by a pre-defined error correction coding scheme. Finally the content hash value of the image is obtained by concatenating the ECC decoder outputs of all tiles.

However, some limitations also exist for this type of content hash based schemes. Due to a short representation of generated content hash, it is very hard for content hash itself to differentiate incidental distortions from intentional distortions, especially when the intentional distortions are the results of attacks to only part of the image. Consequently, it is very difficult to set a proper threshold for making the authentication decision. Furthermore, this scheme also lacks the ability to locate the content modifications if the authentication fails.

We can summarize that, in order to both protect content integrity and prevent sender's repudiation, a good semi-fragile digital signature scheme should satisfy the following requirements. Firstly, the size of the signature signed by a semi-fragile scheme should be small enough to reduce the computational complexity. Ideally, the size is comparable to or the same as that which is signed by a fragile signature scheme (i.e., traditional crypto signature scheme). Secondly, such a scheme should be able to locate the attacked portion of the image because that would easily prove the authentication result. Lastly and most importantly, a proper content-based invariant feature measure should be selected in such a way that it is sensitive to malicious attacks (intentional distortions) and robust against acceptable manipulations (incidental distortions). In addition, the feature should be able to characterize the local property of an image because most attacks only act on part of the image (e.g., only changing the digits in a cheque image not the whole image); Feature selection is obviously application dependent since different applications have different definitions of incidental distortions as well as intentional distortions. Therefore, defining acceptable manipulations and unallowable modifications of the image content is the first step in designing a good semi-fragile authentication system.

III. INCIDENTAL DISTORTIONS AND INTENTIONAL DISTORTIONS

As we described in the previous section, a properly selected feature is very important for system security as well as robustness. A good feature should be sensitive to any intentional distortions while insensitive to all incidental distortions. However, feature selection is application dependent. A good feature for one application isn't necessarily good for another. In other words, only after the application is specified we can analyze which content manipulations are acceptable and which content modifications are not allowable. In this paper, we identify a list of acceptable manipulations for our proposed solution, which may not be complete.

- **Recompression**—Typically this involves re-encoding a compressed image to a lower quality and the re-encoding may be repeated multiple times. In practice, there is a minimal requirement for image quality, which is usually represented by JPEG quantization step size. Thus, one may set a maximum acceptable quantization step size beyond which the image will be considered unauthentic. Such acceptable manipulations will unavoidably introduce some incidental distortions.
- **Format Conversion** – Image differences may exist between its JPEG format and other formats such as BMP. Such differences can be due to different accuracies of representation in the domains of spatial as well as transformation. Since image format conversion occurs very often during image editing, storage, and display, it has to be deemed as another acceptable manipulation.
- **Watermarking** – Image data is "manipulated" when authentication feature codes are embedded back into the image. Such a manipulation should not cause the resulting image to be unauthentic, which means it

should be considered as acceptable. Usually it requires the authentication scheme to be robust to random noise as the distribution of most watermarks could be modeled as random noise such as Gaussion.

Figure 2 further shows the images with different types of distortions. Figure 2(a) is the original "Lena" image with the size of 512x512. Figure 2(b) is the JPEG compressed image with the quality factor 30. Although the difference between the original image and its compressed image is visible, such distortion has to be deemed as acceptable because it comes from the procedure of JPEG lossy compression. Similarly Figure 2(c) corrupted by Gaussion noise has also to be deemed as acceptable because most watermarks are noise-like. However, Figure 2(d) should be considered as an attacked image because the manipulation done may result in a misinterpretation of image meaning ---- adding texts in up-right corner and white spots on Lena's hair.

It is more interesting when we project the image distortions shown in Figure 2 onto DCT domain instead of spatial domain [20], as shown in Figure 3. The distortions are illustrated in terms of the difference of corresponding DCT values between the original image (Horizontal Axis) and the image to be tested (Vertical Axis). Therefore, all plots should lie on the line of 45^o angle if no any difference exists between the original image and the image to be tested. Figure 3(a) and (b) are the distortions from JPEG compression and Figure 3(c) and (d) are the distortions from additive noise: all are acceptable. The distortions are always distributed along the line of 45^o angle: the stronger the distortion is the farther it is to the line 45^o angle. Figure 3(e) and (f) are the distortions from attacks: all are malicious. We observed that the distortion distribution of the acceptable manipulations could be bounded while the distortion distribution of malicious attacks is scattered over the whole plane. We could then draw two lines with 45^o angle to differentiate the acceptable manipulations from malicious attacks, as shown in Figure 4. However, such boundary is not a hard boundary which is assumed in [21]. Such observation further supports our earlier statement on the reason why we cannot directly apply traditional crypto signature scheme for robust image authentication.

To tackle the issue of authenticating the message with soft boundary, we have proposed a novel solution by incorporating error correction coding [22] and watermarking into the traditional crypto signature scheme [1, 2, 3]. In next Section we shall describe a JPEG compliant media signature scheme based on our previous work but with different ECC and watermarking strategies.

IV. Proposed JPEG-compliant media signature scheme

A typical JPEG compression procedure includes block formation, DCT, quantization and lossless entropy coding. In this paper, we select DCT coefficient as the feature. Denote a DCT coefficient before quantization as *D*, the quantization step size specified in the quantization table as *Q*, and the output of quantizer as quotient *F* (integer rounding) and remainder *R* respectively. We have

$$D / Q = F, D\% Q = R = D - F * Q \qquad (1)$$

For JPEG compression, the *F* will be losslessly compressed and *R* will be discarded. Suppose the incidental distortion introduced by acceptable manipulations can be modeled as noise whose maximum absolute magnitude is denoted as *N*, we can then use *R* to correct the errors of *F* caused by corruption from added noise. Refer to Figure 5, assuming $Q > 4N$, *N* is the maximum range of added noise, we can see that if the original DCT value is located at the point nQ, then no matter how this value is corrupted, the distorted value will still be in the range $((n-0.5)Q,(n+0.5)Q)$, and the quantized DCT value will remain unchanged as nQ before and after noise addition. However, if the original DCT value drops into the range of $((n-0.5)Q, nQ)$ (the point *P* in Figure 5, its quantized value is still nQ before adding noise, but there is also a possibility that the noisy DCT value could drop at the range$((n-1)Q,(n-0.5)Q)$ and will be quantized as $(n-1)Q$, not *nQ,* after adding noise. Thus the noise corruption will cause a different quantization result. To avoid such a case, we propose a simple ECC-like procedure to record the sign of *R*. We want to push the points away from the quantization decision boundaries and create a margin of at least *Q*/4 so that the DCT value when contaminated later will not exceed the quantization decision boundaries.

In the ECC procedure, let's record a 0 bit if the original DCT value drops between $((n-0.5)Q, nQ)$ (i.e., $R < 0$). In the authentication procedure, assume this DCT value has been corrupted. Add the value 0.25*Q* from the corrupted value before quantizing it if we retrieve a 0 bit indicating R<0. Then we can obtain the same quantization value as nQ. Similarly, for the case that the original DCT value is in $(nQ,(n+0.5)Q)$ (i.e., $R > 0$), we record a 1 bit and we should subtract 0.25*Q* from the corrupted DCT value before the quantization. We can still obtain the same quantized value as nQ. Based on such an error correction concept, the crypto hashed value on all quantized DCT values will stay the same before and after distortion. This forms the basis of our solution.

In [23], the authors presented a content-based watermarking solution for JPEG image authentication. Two quantization step sizes are used: Q_{a} is for generating features and watermarking while Q_{c} is for actual JPEG compression. They proved that as long as Q_{c} is less than than Q_{a}, the robustness of generated features as well as embedded watermarks can be guaranteed. We shall use this concept in our solution. The whole media signature signing/verification algorithm is depicted as follows and shown in Figure 6 and 7 respectively.

Signature Generation

Initialization

Random sequence *S* for feature selection / watermarking (*S* is used to decide which DCT values are used for feature extraction and which are used for watermarking and it will be included in the signature for verification purpose)

Input

Image owner's private key *Pri*.

Original image I_o to be protected.

Authentication quantization step size $Q_{\text{a.}}$

JPEG compression quantization step size Q_c. Note $Q_c < Q_a$

Begin

Normal JPEG compression processing such as blocking, DCT, obtaining a number of 8x8 DCT blocks in zig-zag scan denoted as: $\{D^O_{ij} : 0 \le i < 64;\ 0 \le j < N\}$.

For $j = 0 : N - 1$ **Do**

Take DC and other 3 AC coefficients randomly selected by *S* and form feature set *F* containing 4 elements:

$$F_j = \left\{D^O_{0j}; D^O_{lj} : l = \{i : 1 \le i \le 20\}^{(3)}_S\right\}.$$

Compute rounding quantization on F_j, obtain quantized value $\overline{F_j}$ and remainders R_j according to (1) with Q_a.

Generate watermark W_j with $W_j = \begin{cases} 0, & if\ R_j < 0 \\ 1, & if\ R_j \geq 0 \end{cases}$.

Embed W_j into the same block by using other AC coefs labeling $1 \leq i \leq 20$ but excluding those which have been used for feature set, refer to [23] for detailed watermarking scheme.

De-quantize all processed DCT coeffs with $Q_{a.}$

End

Crypto hash all concatenated $\overline{F}_j$: $H^O = h(\cup \overline{F}_j)$, $0 \leq j < N$.

Sign on H^O by *Pri* and obtain signature *G*.

Compress watermarked image I_w with Q_c quality (i.e., quantize all DCT coeffs with Q_c) and obtain I_w.

End

Output

Compressed watermarked image I_w.

Content based signature *G.*

Signature Verification

Same random sequence *S* obtained from the signature.

Input

Image I_w to be authenticated.

Owner's public key *Pub*

Associated signature *G*.

Authentication quantization step size $Q_{a.}$

JPEG compression quantization step size Q_c. Note $Q_c < Q_a$

Begin

Normal JPEG decoding such as Huffman, de-quantizing with Q_c, obtaining a number of 8x8 DCT blocks in zig-zag scan denoted as: $\{D_{ij}^W : 0 \leq i < 64;\ 0 \leq j < N\}$.

For $j = 0 : N - 1$ **Do**

Take DC and other 3 AC coefficients randomly selected by *S* and form feature set F^W containing 4 elements:

$$F_j^W = \left\{ D_{0j}^W ; D_{lj}^W : l = \{ i : 1 \le i \le 20 \}_S^{(3)} \right\}.$$

Extract W_j from the same block, refer to [23] for detailed watermark extraction procedure.

Modify F_j^W according to the extracted W_j:

$$F_j^W = \begin{cases} F_j^W + 0.25 Q_a, & if\; W_j = 0 \\ F_j^W - 0.25 Q_a, & if\; W_j = 1 \end{cases}$$

Compute rounding quantization on F_j^W and obtain quantized values $\overline{F}_j^W$ and remainders R_j^W according to (1) with Q_a.

Correlate W_j with R_j^W , obtain difference image I_d.

End

Crypto hash all concatenated $\overline{F}_j^W : H^W = h(\cup \overline{F}_j^W)$, $0 \le j < N$.

Decrypt signature *G* by owner's public key *Pub,* obtain hash H^O.

Bit-wise comparison between H^O and H^w: $A = xor\left(H^O, H^W\right)$

End

Output

If $A > 1$, then report image unauthentic and display I_d to indicate possible attacked locations.

If $A = 0$, the image is authentic.

Note that sometimes difference image I_d fails to indicate the modification locations although the image is verified as unauthentic based on bit-wise comparison between H_o and H_w. For example, DCT values originally closed to $n*Q$ will be pushed to the opposite side and thus the sign changes due to acceptable manipulations. Or similarly, when there are attacks, there might not be sign changes, e.g., the change to the DCT value is large, pushing the value from "-" side of $n*Q$ to "-" side of $(n+1)*Q$, even after $Q/4$ adjustment. However, the crypto hash value definitely will be changed, because the quantized values are changed.

V. System performance analysis and evaluation

1. System security analysis

In this section, firstly we investigate the security of our media signature solution. Since the whole media signature scheme is compliant with crypto digital signature, we only need to analyze the security of three main parts, namely: feature extraction, ECC and crypto hash, as shown in Figure 8(a). Therefore, if we denote the system security in terms of the possibility of being attacked (i.e., the possibility of finding a faked image which can pass the authentication test), the security for our scheme should comprise of three possibilities: P_F in feature extraction, P_E in error correction coding and P_C in crypto hashing. (The security for crypto hashing function (e.g., 160 bits in SHA-1) is [5]: $P_C \approx 2^{-80}$ under the well-known "birthday attack"). Since they are mutually independent and very small, the final security could be approximated as: $P = 1 - (1 - P_F)(1 - P_E)(1 - P_C) \approx P_F + P_E + P_C$. Obviously it is much larger than P_C, which is usually deemed as nearly perfectly secure based on current computational capabilities. Therefore, we only need to study P_F and P_E which impair the system security in different ways, as shown in Figure 8(a). A good feature descriptor should represent as much entropy of the original source as possible. Differing from feature extraction which functions as "removing" redundancy from original source, ECC functions as "adding" redundancy in order to tolerate incidental distortions. Thus, a good feature extraction method and a proper ECC scheme are the key factors in system security. Let's check P_E first. Originated from [22], we have:

Lemma: Let H_C be our proposed media hash scheme based on an ECC with error correction ability t (e.g., $t = 0.25Q$). For any D' which satisfies $\|D - D'\| \le t$, we have $H_C(F) = H_C(F')$.

Note that the notations in the above lemma are the same as our described media signature algorithm. We skip its proof here due to the paper size limit. Basically, it states that as long as the difference between original DCT values and corrupted DCT values are not greater than *t*, their corresponding hash values are the same as by using ECC. Clearly, ECC diminishes system security in some sense as ECC does provide the property of fuzziness. This goes back to the issue of how to select a proper ECC scheme to balance between system robustness and security. This issue is application dependent. In our proposed scheme for JPEG (refer to the algorithm described above), we take the remainder of quantization of DCT coefficient for error correction while hashing quantized DCT values. Since the maximum magnitude of the remainder is less than half of quantization step size Q_a and it is usually discarded in JPEG compression, we argue that the security risk caused by this particular ECC scheme should be ignored. We mean that changing and attacking on remainder of DCT values will not affect the system security while attacking quantized DCT values is computationally impossible because a crypto hashing works on protecting all quantized DCT values.

Before analysing P_F , we make the following assumption:

Assumption: Lossy compression (e.g, JPEG) is the approximate representation of its original source in terms of an optimal rate-distortion measure.

Intuitively, it implies there is no security gap between the original source and its compressed version under a given targeted bit-rate. Therefore, we can argue that if a media hashing function could make use of all compressed information (e.g., all quantized DCT coefficients in all blocks), it should be considered secure at this targeted bit rate. In other words, if an attacker intends to modify the content in the spaces between original source and its compressed version, this attack should not be deemed harmful to the content, as shown in Figure 8(b) (The shading range between Q_a and Q_c). Therefore, we suggest to set Q_a and Q_c as close as possible to reduce the space for attackers.

In practice, we may have to pick up only some DCT coefficients for hash generation because we need to store ECC check information back into the content. In such a case, the system security will suffer. Since the low frequency DCT coefficients are more important than high frequency DCT coefficients, selecting DCT coefficients from a low frequency band for hash generation, as we did in our proposed scheme, will gain more security than selecting from a high frequency band. The worst security performance of our proposed scheme can then be estimated by simply assuming all DCT coefficients are of the same importance to system security. As we described before, we select one DC and three AC components for feature formation and embed them back into other AC components again, all in the range of the first 20 DCT coefficients which are from the low to middle frequency band. Therefore, analyzing P_F in one 8x8 block is equivalent to computing the possibility in finding 7 balls (3 from AC features and 4 for embedding) with a fixed order from 20 balls: $P_F = (20-7)!/20! \approx 2.56 \times 10^{-9}$. This represents the probability of the attacker guessing correctly the locations of the 7 coefficients in an exhaustive attempt. If he can determine the correct locations, he can then keep the DCT values at these locations unchanged, and try to cause harmful misunderstanding by exhaustively changing other DCT values in this block. Although it is still very difficult, especially when we take the contextual property (i.e., the values of its surrounding pixels are very similar to the central one) of image into account, we have to analyze the worst case scenario from the viewpoint of system security. The final security could then be: $P = P_F + P_E + P_C \approx P_F = (2.56 \times 10^{-9})^N$, where N is the total number of blocks in an image. We can see P is not so high from the viewpoint of cryptography, especially in the case of attacking locally (i.e., N may be 1). However, such security performance should be able to satisfy the requirements of real applications, considering that strong contextual property in image will also increase the difficulties in attacking.

The last issue related to security is watermarking. In our proposed scheme, we do not need to pay more attention to watermarking security except by using *S* to randomly select watermarking locations. The main reason is that watermarking here only functions to store ECC check information. Since the distribution of remainder *R* is independent from $\overline{F}$, and it does not reveal any information about $\overline{F}$, leaving *R* in the public will not affect the security of hash function [22].

2. Experimental results

We used 10 images for our system test, as shown in Figure 9. They are of different sizes varying from 256x256 to 720x576. In our proposed scheme, we simply define the maximum allowable distortions as acceptable manipulations which is set to 1/4 of the quantization step size. RSA is used to sign and verify the generated media signature whose length is 1024 bits. Figure 10 shows the examples of the results of image authentication. Figure 10(a) is the noisy watermarked image with an additive noise of zero mean and 0.01

variance. This noisy image can be verified successfully. Figure 10(b) is the attacked watermarked image (the window in the center was removed and was filled with its surrounding content). The authentication fails and the result of locating attacks is shown in Figure 10(c). We also adopted some image pre-processing techniques such as low pass filtering and histogram equalization to help to stabilize signature verification [18]. As an implementation issue, the watermarking method we adopted is the same as [23] except for the blocks where watermarks cannot be embedded due to visual quality consideration. For those blocks, we simply substitute a random sequence generated by S and block location for the features extracted from this block and use it for crypto hashing. More extensive experiments will be conducted as our future work both in security tests as well as robustness tests.

The evaluation on the quality of watermarked images is shown in Figure 11. The authentication quantization size and the compression quantization step size are both set to the JPEG compression quality factor 75. The upper curve is the PSNR values of the compressed image without watermarking (i.e., purely compressed by JPEG) and the lower curve is the PSNR values of the compressed image with watermarking (i.e., both JPEG compressed and watermarked), all are with the reference to their corresponding original images. The average quality loss due to watermarking is about 1.2 dB. But no significant visual difference between these two sets of images.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a new semi-fragile image authentication solution combining ECC and watermarking. By using ECC, we provide a mechanism allowing minor variations of content features caused by acceptable manipulations (such as lossy compression and additive noise). In summary, our media signature scheme not only eliminates the signature size issue in previous signature-based methods but also eliminates the security issue in previous watermarking based methods. The whole solution is compliant with Public Key Infrastructure (PKI) while retaining promising system robustness as well as security performance. Future work includes more rigorous testing and analysis and extending this process to other media such as audio and video.

VII. ACKNOWLEDGMENT

The authors would like to thank Dr. Ching-Yung Lin (IBM T.J. Waston) for sharing many thoughtful views on media authentication and two students Mr. Shuiming Ye and Mr. Xinkai Wang for system implementation.

VIII. REFERENCES

[1] Q. Sun, S.-F. Chang, K. Maeno and M. Suto, "A new semi-fragile image authentication framework combining ECC and PKI infrastructure", *International Symposium on Circuits and Systems (ISCAS02)*, Phoneix, USA, 2002.

[2] Q. Sun, S.-F. Chang, M. Kurato and M. Suto, "A quantitive semi-fragile JPEG2000 image authentication system", *International Conference on Image Processing (ICIP02)*, Rochester, USA, 2002.

[3] Q. Sun and S.-F. Chang, "Semi-fragile image authentication using generic wavelet domain features and ECC", *International Conference on Image Processing (ICIP02)*, Rochester, USA, 2002.

[4] P. W. Wong and N. Memon, "Secret and public image watermarking schemes for image authentication and ownership verification", *IEEE Transactions on Image Processing*, Vol.10, No.10, pp.1593-1601, 2001.

[5] M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification", *International Conference on Image Processing (ICIP97)*, Santa Barbara, USA, 1997.

[6] B. Schneier, *Applied Cryptography*, New York: Wiley, 1996

[7] G. L. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image", *IEEE Transactions on Consumer Electronics*, Vol.39, No.4, pp.905-910, 1993.

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

[9] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication", *International Conference on Image Processing (ICIP98)*, Chicago, USA, 1998.

[10] M. Wu and B. Liu, "Watermarking for image authentication", *International Conference on Image Processing (ICIP98)*, Chicago, USA, 1998.

[11] J. Fridrich, "Image watermarking for tamper detection", *International Conference on Image Processing (ICIP98)*, Chicago, USA, 1998.

[12] M. Schneider and S.-F. Chang, A robust content-based digital signature for image authentication, *International Conference on Image Processing (ICIP96)*, 1996.

[13] C.-Y. Lin and S.-F. Chang, A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.11, No.2, pp.153-168, Feb. 2001

[14] C.-S. Lu and H.-Y. Mark Liao, "Structural Digital Signature for Image Authentication: An Incidental Distortion Resistant Scheme", *IEEE Trans. on Multimedia*, Vol. 5, No. 2, pp. 161-173, 2003.

[15] M. P. Queluz, "Content Based Integrity Protection of Digital Images", *SPIE International Conf. on Security and Watermarking of Multimedia Contents*, vol. 3657, No. 09, EI '99, San Jose, USA, Jan 1999.

[16] D. Johnson and A. Menezes, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Univ. of Waterloo, 1999, appeared in http://citeseer.nj.nec. com/johnson99elliptic.html

[17] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking", *Proceedings of IEEE International Conference on Information Technology - Coding and Computing'00*, Las Vegas, March, 2000.

[18] L. Xie, G. R. Arce and R. F. Graveman, "Approximate image message authentication codes", *IEEE Transactions on Multimedia*, Vol.3, No.2, pp.242-252, 2001.

[19] R. Venkatesan, S.-M. Koon, M. H. Jakubowski and P. Moulin, “Robust image hashing”, *International Conference on Image Processing’00*, Vancouver, Canada, Sept, 2000.

[20] X. Wang, “Robust digital signature for image authentication”, *Master Thesis*, National University of Singapore, 2000.

[21] N. Memon, P. Vora, B. Yeo and M. Yeung, “Distortion bounded authentication techniques”, *Proceedings of the SPIE, Security and Watermarking of Multimedia Content II*, Vol.3971, pp.164-174, 2000.

[22] A Juels and M. Wattenberg, “A fuzzy commitment scheme”, *Proceedings of ACM Conference on Computer and Communications Security’99*, Singapore, 1999.

[23] C.-Y. Lin and S.-F. Chang, “Semi-Fragile Watermarking for Authenticating JPEG Visual Content”, *SPIE Security and Watermarking of Multimedia Contents II EI '00*, SanJose, CA, Jan. 2000

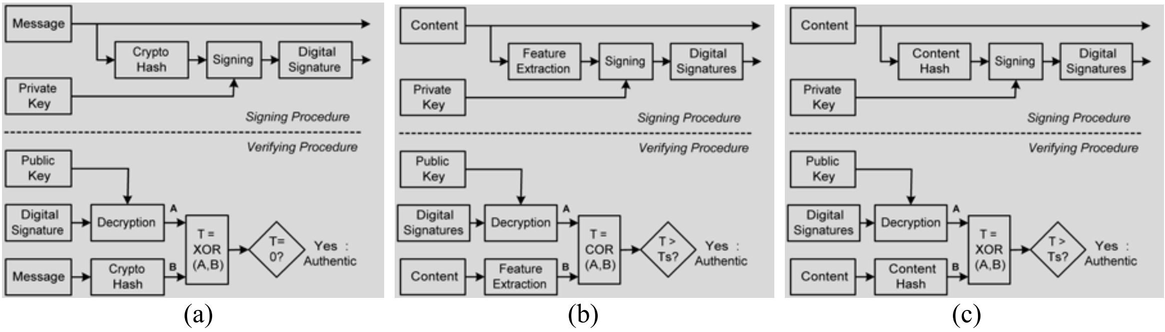


Figure 1. Digital signature schemes for message / content authentication. (a) Traditional crypto signature scheme for fragile authentication. (b) Feature based crypto signature scheme for semi-fragile authentication. (c) Robust hash based crypto signature scheme for semi-fragile authentication.

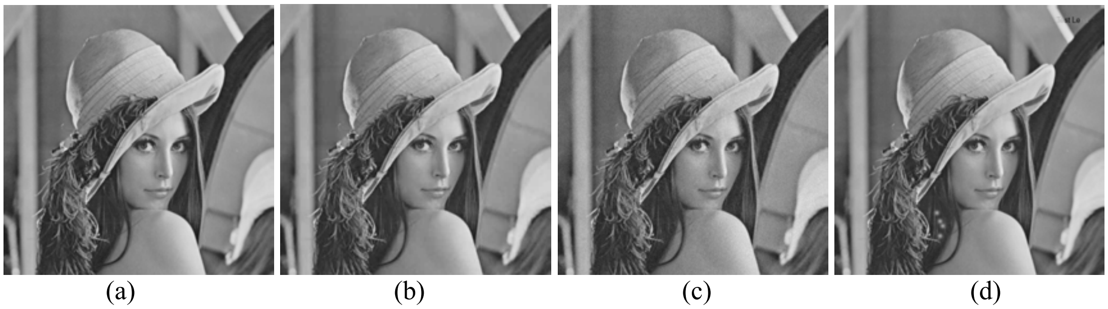
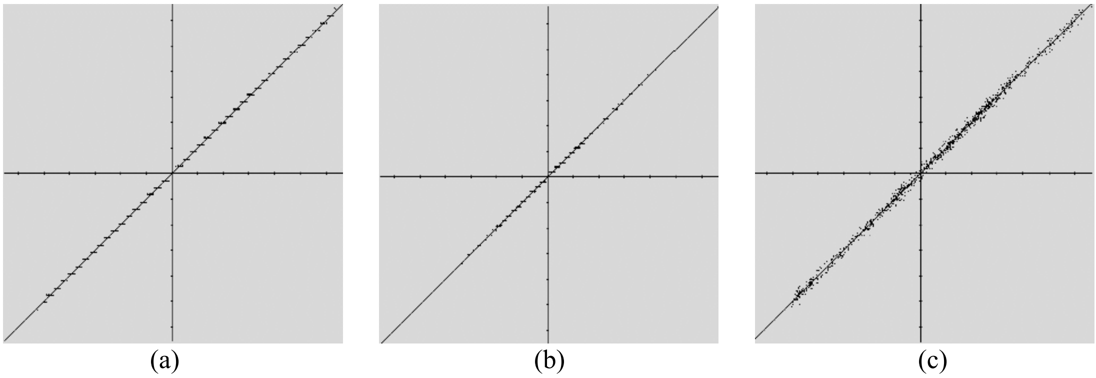


Figure 2. (a) Original image “Lena” with size 512x512 (b) JPEG compressed image with quality factor 30 (c) Noise corrupted image with additive strength 5% (d) Attacked image.



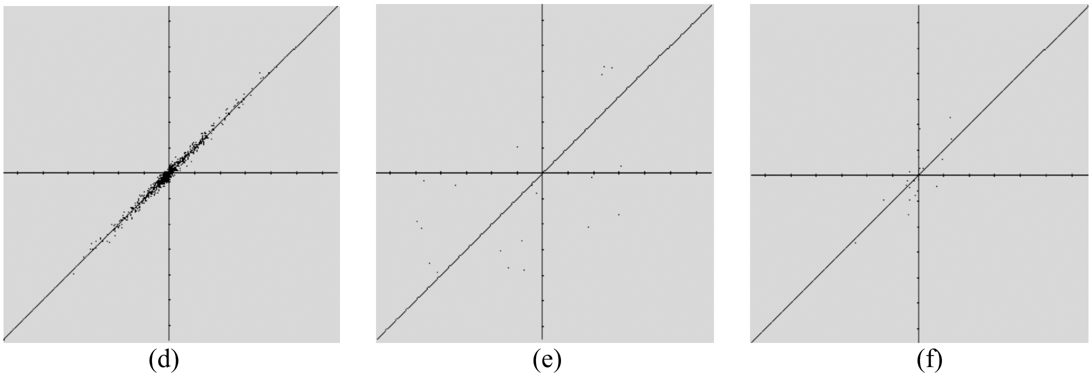


Figure 3. Distortion illustration in terms of the difference of DCT value between the original image and the testing image. All points should lie on the line with 45° if no any difference between those two images. (a) With JPEG compressed image: DC component. (b) With JPEG compressed image: 1st AC component. (c) With noise corrupted image: DC component. (d) With noise corrupted image: 1st AC component. (e) With attacked image: DC component. (f) With attacked image: 1st AC component.

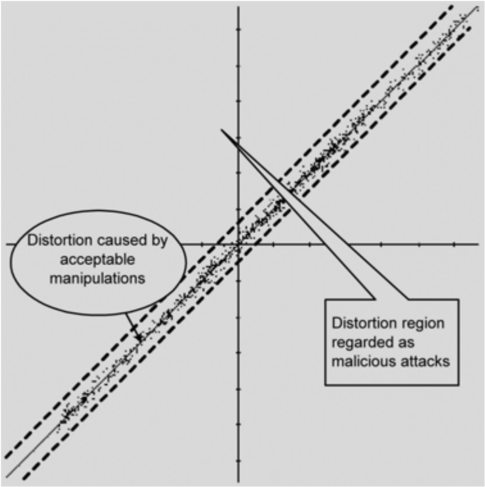


Figure 4. Idea illustration of Distortion "bounded" authentication. We observed that such bound is soft not hard which means it is not applicable to directly adopt crypto hashing to image content.

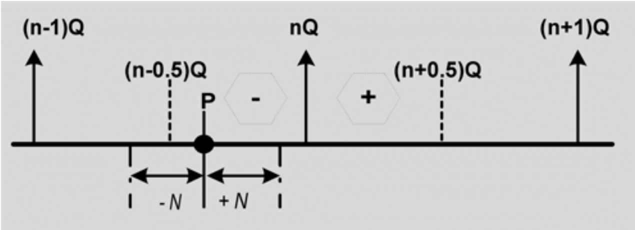


Figure 5. Illustration on the concept of error correction for robust authentcation

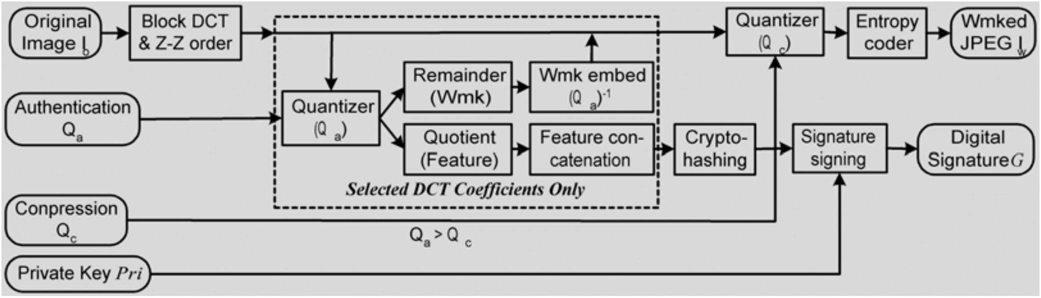


Figure 6. Diagram of the proposed semi-fragile authentication system (Image signing)

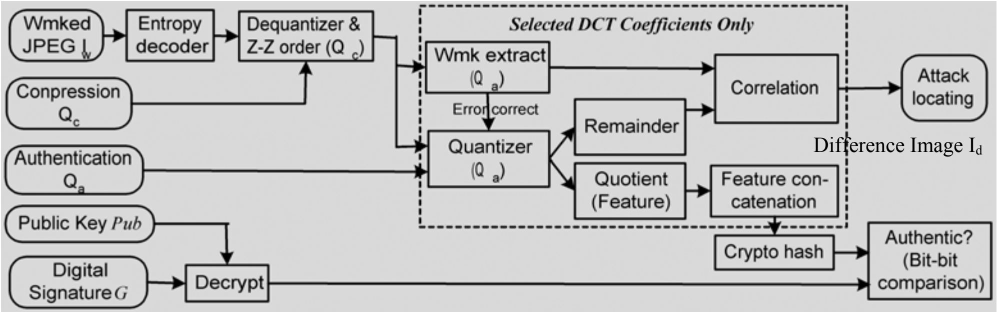
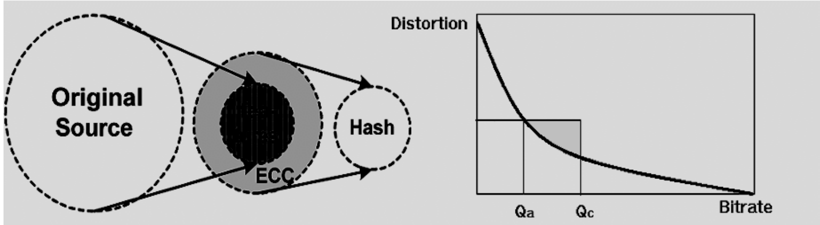
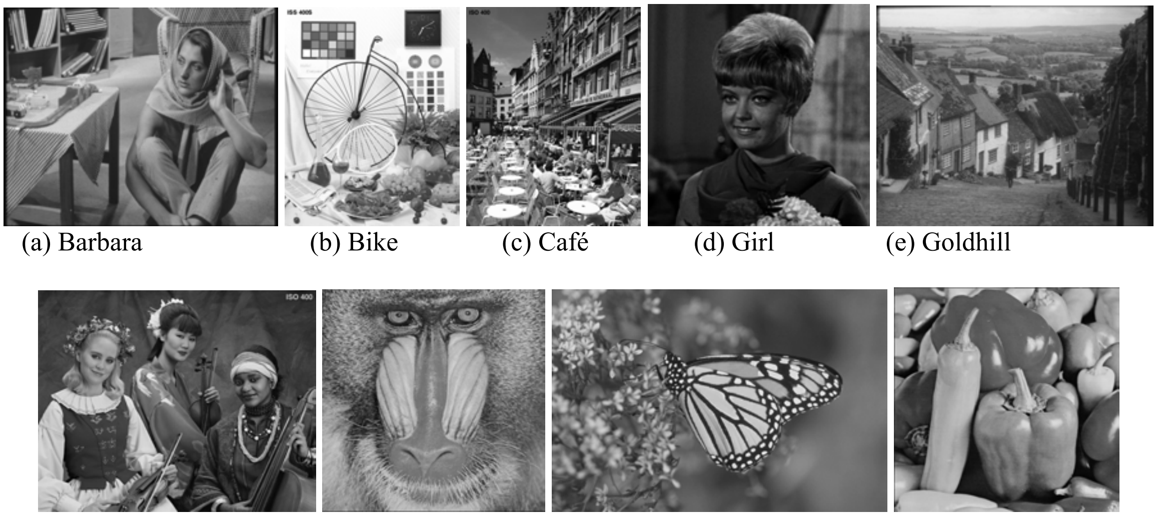


Figure 7. Diagram of the proposed semi-fragile authentication system (Image verifying)



(a) The flow of the signature formation (b) The range of authentic distortion region

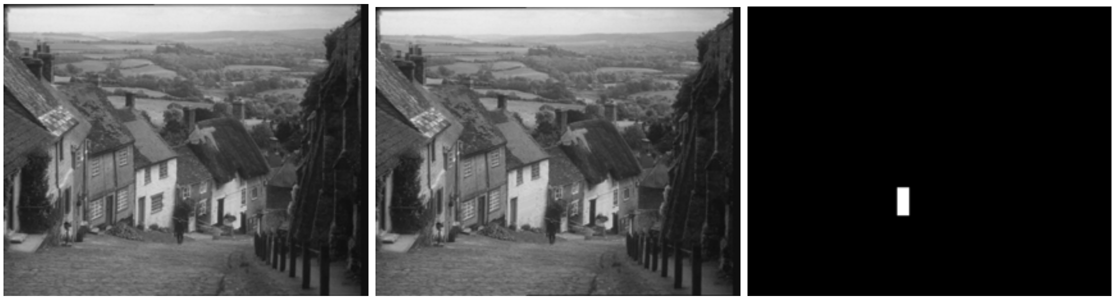
Figure 8. Illustration for the security analysis



(a) Barbara (b) Bike (c) Café (d) Girl (e) Goldhill

(f) Women (g) Mandrill (h) Monarch (i) Peppers

Figure 9. Images used for testing



(a) Image with additive noise (b) Attacked image (A window is removed) (c) Attack locating

Figure 10. Examples of authentication results

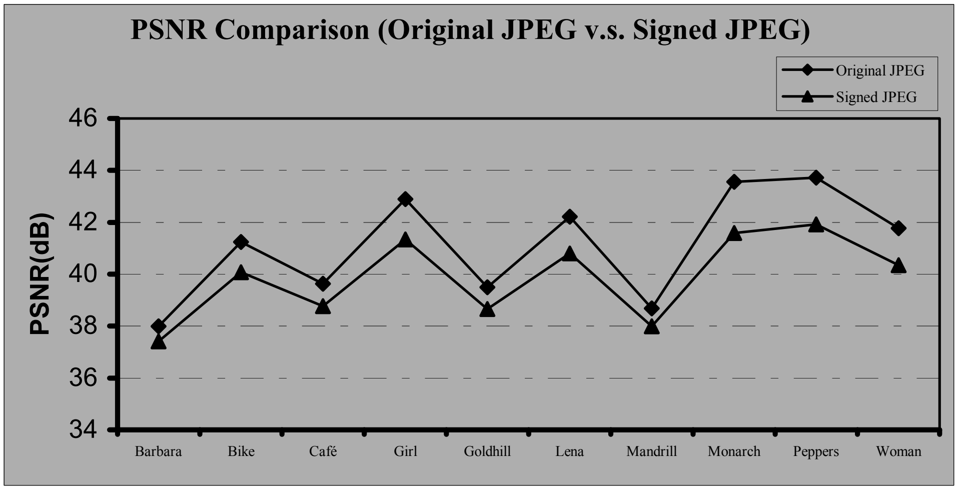


Figure 11. Quality evaluation of watermarked images.