Robust Watermarking of Compressed and Encrypted JPEG2000 Images

A. V. Subramanyam, Sabu Emmanuel, *Member, IEEE*, and Mohan S. Kankanhalli, *Senior Member, IEEE*

***Abstract*—Digital asset management systems (DAMS) generally handle media data in a compressed and encrypted form. It is sometimes necessary to watermark these compressed encrypted media items in the compressed-encrypted domain itself for tamper detection or ownership declaration or copyright management purposes. It is a challenge to watermark these compressed encrypted streams as the compression process would have packed the information of raw media into a low number of bits and encryption would have randomized the compressed bit stream. Attempting to watermark such a randomized bit stream can cause a dramatic degradation of the media quality. Thus it is necessary to choose an encryption scheme that is both secure and will allow watermarking in a predictable manner in the compressed encrypted domain. In this paper, we propose a robust watermarking algorithm to watermark JPEG2000 compressed and encrypted images. The encryption algorithm we propose to use is a stream cipher. While the proposed technique embeds watermark in the compressed-encrypted domain, the extraction of watermark can be done in the decrypted domain. We investigate in detail the embedding capacity, robustness, perceptual quality and security of the proposed algorithm, using these watermarking schemes: Spread Spectrum (SS), Scalar Costa Scheme Quantization Index Modulation (SCS-QIM), and Rational Dither Modulation (RDM).**

***Index Terms*—Compressed and encrypted domain watermarking, JPEG2000.**

I. Introduction

DIGITAL media content creation/capturing, processing and distribution has witnessed a phenomenal growth over the past decade. This media content is often distributed in compressed and encrypted format and watermarking of these media for copyright violation detection, proof of ownership or distributorship, media authentication, sometimes need to be carried out in compressed-encrypted domain. One such example is the distribution through DRM systems [1]–[4] where the owner of multimedia content, distribute it in a compressed and encrypted format to consumers through multilevel distributor network. In DRM systems with content owners, multiple levels of distributors and consumers, the distributors do not have access to plain content (un-encrypted content). As they are distributors of content who distributes the encrypted content (in fact compressed encrypted content as most of the content would be compressed and then encrypted) and requests the license server in the DRM system to distribute the associated licence containing the decryption keys to open the encrypted content to the consumers. In fact distributors do not need to have plain content as they are not consumers. However, each distributor sometime needs to watermark the content for media authentication, traitor tracing or proving the distributorship. Thus they have no choice but to watermark in the compressed encrypted domain.

In this paper we focus on watermarking of compressed-encrypted JPEG2000 images, where the encryption refers to the ciphering of complete JPEG2000 compressed stream except headers and marker segments, which are left in plaintext for format compliance [5]. There have been several related image watermarking techniques proposed to date [6]–[11]. In [6], Deng *et al.* proposed an efficient buyer-seller watermarking protocol based on composite signal representation given in [7]. However, when the content is accessible only in encrypted form to the watermark embedder, the embedding scheme proposed in [6] might not be applicable as the host and watermark signal are represented in composite signal form using the plain text features of the host signal and in [6], this is possible as the seller embeds the watermark. Also, there is a ciphertext expansion of 3.7 times that of plaintext. In [8] and [9], some sub-bands of lower resolutions are chosen for encryption while watermarking the rest of higher resolution sub-bands. While in [10], the encryption is performed on most significant bit planes while watermarking the rest of lower significant bit planes. In case lesser number of sub-bands/bit planes are used for encryption, an attacker can manipulate the un-encrypted sub-bands/bit planes and further extract some useful information from the image, although the image may not be of good quality. On the other hand, if more sub-bands/bit planes are encrypted and only rest few sub-bands/bit planes are watermarked, it might be possible for an attacker to remove the watermarked sub-bands/bit planes while maintaining the image quality. Prins *et al.* in [11] proposed a robust quantization index modulation (QIM) based watermarking technique, which embeds the watermark in the encrypted domain. In the technique proposed in [11], the addition or subtraction of a watermark bit to a sample is based on the value of quantized plaintext sample. However, in our algorithm, the watermark embedder does not have access to the plain text values. They have only compressed-encrypted content. Also the watermark embedders do not have the key to un-encrypt and get the plain text compressed values. Thus, watermarking in compressed-encrypted domain using the

Manuscript received September 26, 2010; revised March 05, 2011, August 10, 2011, and November 11, 2011; accepted November 29, 2011. Date of publication December 23, 2011; date of current version May 11, 2012. This work was supported by the Agency for Science, Technology and Research (A*STAR), Singapore, under the project "Digital Rights Violation Detection for Digital Asset Management" (Project No: 0721010022). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Oscar C. Au.

A. V. Subramanyam and S. Emmanuel are with the School of Computer Engineering, Nanyang Technological University, Singapore (e-mail: subr0021@ntu.edu.sg; asemmanuel@ntu.edu.sg).

M. S. Kankanhalli is with the School of Computing, National University of Singapore, Singapore (e-mail: mohan@comp.nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2181342

technique proposed in [11] is very challenging. In [12] Li *et al.* proposed a content-dependent watermarking technique, which embeds the watermark in an encrypted format, but the host signal is still in the plain text format. The algorithm may not be directly applied when the content is in the encrypted format, in which case the distortion introduced in the host signal may be large. In [13] Sun *et al.* proposed a semi fragile authentication system for JPEG2000 images. However, this scheme is not fully compressed and encrypted domain watermarking compatible as it derives the content based features for watermarking from the plain text.

We propose a robust watermarking technique for JPEG2000 images in which the watermark can be embedded in a predictable manner in compressed-encrypted bytestream by exploiting the homomorphic property, explained in Section II-A, of the cipher scheme. Watermarking in compressed-encrypted content saves the computational complexity as it does not require decompression or decryption, and also preserves the confidentiality of the content. However, this proposed technique faces the following challenges.

1) Compressed Domain Watermarking: A small modification in the compressed data may lead to a considerable deterioration in the quality of decoded image. Thus the position for watermark embedding has to be carefully identified in the compressed data, so that the degradation in the perceptual quality of image is minimal.

2) Encrypted Domain Watermarking and Watermark Retrieval: In an encrypted piece of content, changing even a single bit may lead to a random decryption, therefore the encryption should be such that the distortion due to embedding can be controlled to maintain the image quality. It should also be possible to detect the watermark correctly even after the content is decrypted. Also, the compression gain should not be lost as encryption may lead to cipher text expansion.

Although, some asymmetric schemes like RSA [14], Goldwasser-Micali [15], Elgamel [16] and Paillier [17], with homomorphic property, can be used for encryption purpose, there are two main drawbacks using such schemes. Firstly, if the encryption is performed on a message size of few bits, the size of the ciphertext may expand leading to loss of compression efficiency. For RSA and Goldwasser-Micali, expansion is caused due to the use of modulo $n_{p'q'}$ (a product of two large primes p' and q'). For Paillier and Elgamel the expansion of ciphertext to plaintext is 2 [17], [16]. Secondly, if the encryption is performed on a large message size, say, few hundreds of bits, to compensate the loss in compression, the payload capacity decreases, where payload capacity is the number of watermark signal bits embedded per encrypted message. A secure symmetric stream cipher with homomorphic property is preferred over secure asymmetric encryption with homomorphic property mainly due to the following two reasons. Symmetric ciphers with homomorphism can be applied on a smaller message size, like a byte, without increasing the compressed data size and achieving a better payload capacity than asymmetric counterparts. So there is a tradeoff between security-compression efficiency-payload capacity, which poses a challenge for deciding which cipher scheme to apply. Therefore we use the RC4 stream cipher with homomorphism property. This paper is organized as follows. Section II describes the proposed scheme. In Section III, we discuss the key distribution, domain of encryption, security analysis of encryption and watermarking algorithm, and effect of scaling on RDM detection. The experimental results are discussed in Section IV. Section V concludes the paper. The theoretical analysis and derivations are given in the Appendix. Following are the notations which are used in the paper.

Notation:

- L denotes the length in bytes.
- $\boldsymbol{M} = \{m_i\}, m_i \in [0, 255]\ \forall i = 0, 1, \ldots, L-1$ denotes the packetized JPEG2000 bytestream, $\boldsymbol{M_W} = \{m_{w_i}\}, m_{w_i} \in [0, 255]\ \forall i = 0, 1, \ldots, L-1$ to be the watermarked copy of $\boldsymbol{M}$.
- $\boldsymbol{C} = \{c_i\}, c_i \in [0, 255]\ \forall i = 0, 1, \ldots, L-1$ denotes encrypted $\boldsymbol{M}$, $\boldsymbol{C_W} = \{c_{w_i}\}, c_{w_i} \in [0, 255]\ \forall i = 0, 1, \ldots, L-1$ to be the watermarked copy of $\boldsymbol{C}$.
- $\mathbf{b} = \{b_j\}, b_j \in \{-1, 1\}\ \forall i = 0, 1, \ldots, N-1$ denotes the watermark information.
- $\boldsymbol{E}(\cdot)$ and $\boldsymbol{D}(\cdot)$ denotes the encryption and decryption function, respectively.
- $\boldsymbol{K} = \{k_i\}$, where $k_i \in [0, 254]\ \forall i = 0, 1, \ldots, L-1$ denotes the encryption key.
- r denotes the chip rate in SS.
- α denotes the watermark strength factor in SS.
- $\boldsymbol{P} = p_i, p_i \in \{-1, 1\}\ \forall i = 0, 1, \ldots, L-1$ denotes a PN sequence with zero mean and variance σ_p^2.
- K_{qim}, where $k_{qim_i} \in [0, 1)\ \forall i = 0, 1, \ldots, L-1$ denotes a random sequence.
- β denotes the scale parameter in SCS-QIM.
- Δ denotes the step size.
- $Q_\Delta(\cdot)$ denotes scalar uniform quantization, where Δ is the step size.
- $\mathbf{q} = q_i\ \forall i = 0, 1, \ldots, L-1$ denotes the quantization error sequence.
- L_m denotes the number of the past watermarked samples used in RDM.
- γ is the shape parameter in RDM.
- $\mathrm{Exp}(\cdot)$ denotes expectation.
- I and I_w denotes original and watermarked decompressed image, respectively.
- μ_I and σ_I^2 denotes the mean and variance of I, μ_{I_w} and $\sigma_{I_w}^2$ denotes the mean and variance of I_w.
- σ_{II_w} denotes the covariance of I and I_w, c_1 and c_2 are constants.
- $\boldsymbol{N}(0, \sigma_n^2)$ denotes the noise signal with power σ_n^2.
- $\boldsymbol{H_M}$ and $\boldsymbol{H_K}$ denotes the entropy of message $\boldsymbol{M}$ and $\boldsymbol{K}$, respectively.
- $\boldsymbol{P_M}$ and $\boldsymbol{P_{K'}}$ denotes the probability of occurrence of an element in $\boldsymbol{M}$ and $\boldsymbol{K'}$, respectively, where $\boldsymbol{K'}$ is defined similar to $\boldsymbol{K}$.
- $\boldsymbol{P_W}$ and $\boldsymbol{P_N}$ denotes the watermark and noise power, respectively.

II. Proposed Scheme

The proposed algorithm works on JPEG2000 compressed code stream. JPEG2000 compression is divided into five different stages [18]. In the first stage the input image is preprocessed by dividing it into non-overlapping rectangular tiles, the

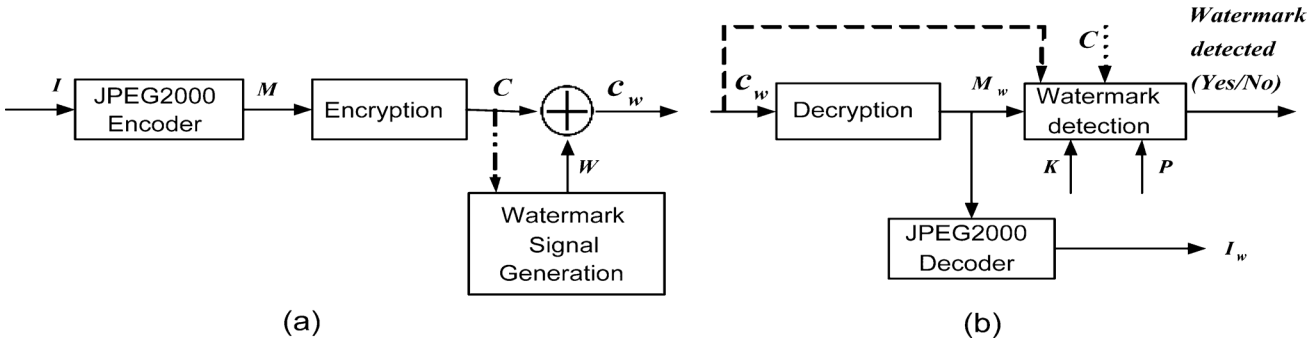


Fig. 1. (a) Watermark embedding (b) Watermark extraction.

unsigned samples are then reduced by a constant to make it symmetric around zero and finally a multi-component transform is performed. In the second stage, the discrete wavelet transform (DWT) is applied followed by quantization in the third stage. Multiple levels of DWT gives a multi-resolution image. The lowest resolution contains the low-pass image while the higher resolutions contains the high-pass image. These resolutions are further divided into smaller blocks known as code-blocks where each code-block is encoded independently. Further, the quantized-DWT coefficients are divided into different bit planes and coded through multiple passes at embedded block coding with optimized truncation (EBCOT) to give compressed byte stream in the fourth stage. The compressed byte stream is arranged into different wavelet packets based on resolution, precincts, components and layers in the fifth and final stage. Thus, it is possible to select bytes generated from different bit planes of different resolutions for encryption and watermarking. The proposed algorithm uses a symmetric stream cipher with additive homomorphic properties for encryption. In fact the distributors get JPEG2000 compressed stream cipher encrypted images for distribution. The distributors can then apply any robust additive watermarking technique to this compressed encrypted stream. In this paper, we use three watermarking schemes, namely, Spread Spectrum (SS), Scalar Costa Scheme (SCS-QIM), and Rational Dither Modulation (RDM) for the purpose and study the bit error rate of detection and the quality versus payload capacity trade-off. Fig. 1 shows the watermark embedding and detection pipelines. The watermark signal for SS is generated without using the host signal, whereas for SCS-QIM and RDM, it is generated using $\boldsymbol{C}$ as shown with dashed-dotted line in the embedding block. The watermark detection can be done before and after the decryption but in the compressed domain as shown in Fig. 1. For detection in encrypted domain, C_W, instead of M_W, is directly fed to the detection module, which is shown in dashed line in Fig. 1. In the watermark extraction block $\boldsymbol{C}$ is given in dotted line to emphasize that it is not required for blind detection. We explain the encryption algorithm next.

A. Encryption Algorithm

JPEG2000 gives out packetized byte stream $\boldsymbol{M}$ as its output. In order to encrypt the message $\boldsymbol{M}$, we choose $\boldsymbol{K}$, a randomly generated key-stream using RC4, explained in Section III-A. Then the encryption is done byte by byte as given in (1) to get the ciphered signal $\boldsymbol{C}$:

$$\begin{aligned}\boldsymbol{C} &= \boldsymbol{E}(\boldsymbol{M}, \boldsymbol{K}) = c_i \\ &= (m_i + k_i) \bmod 255 \ \forall i = 0, 1, \ldots, L-1 \end{aligned} \tag{1}$$

where the addition operation is arithmetic addition. Here, mod 255 is required to preserve the format compliancy of JPEG2000 bit stream [5]. In JPEG2000 bit stream, the header syntax occurs as a value greater than 0xff89. This value correspond to two consecutive bytes having values 255 and higher than 137 in decimal base. If mod 256 is used, it may generate a value 255 and the consecutive byte value greater than 137, which corresponds to a syntax and is undesirable. Thus in order to prevent the generation of header segments, mod 255 is used as given in [5]. Let $\boldsymbol{C}_1 = \boldsymbol{E}(\boldsymbol{M}_1, \boldsymbol{K}_1)$ and $\boldsymbol{C}_2 = \boldsymbol{E}(\boldsymbol{M}_2, \boldsymbol{K}_2)$. For $\boldsymbol{K} = \boldsymbol{K}_1 + \boldsymbol{K}_2$, additive homomorphism property gives

$$\boldsymbol{D}(\boldsymbol{C}_1 + \boldsymbol{C}_2, \boldsymbol{K}) = \boldsymbol{M}_1 + \boldsymbol{M}_2. \tag{2}$$

Here, $M_1 = \{m_{1_i}\}\ \forall i$ has been preprocessed by the owner such that $0 \leq M_1 + M_2 < 255$. The owner does the preprocessing by limiting the values as $M_1 \mid M_1 \in [\alpha, 255 - (\alpha + 1)]$, where α is a positive integer introduced in Section II-B. However, the preprocessing is not applied when $m_{1_i} = 255$ and $m_{1_{i+1}} > 137$, because this case indicates the presence of a header segment which should be preserved to preserve the bitstream compliance. Thus this stream cipher has additive privacy homomorphism property [19]. Since the watermarking technique used is an additive one, the encryption algorithm must have privacy homomorphism property with addition. The privacy homomorphism property will make it possible to detect the watermark from the decrypted content and also help us to control the watermarked image quality easily.

The security of the cryptosystem lies on the underlying stream cipher used. RC4 is a well-established stream cipher and its security has been investigated in depth [23], [24], [26]–[28]. Thus the homomorphic cipher scheme applied here is secure and, further attacks to breach the security of this cipher algorithm are investigated elaborately in Section III-C. Distributors in the distribution chain are given this compressed encrypted byte stream $\boldsymbol{C}$ to distribute. They do not have access to the original image. Often distributors need to watermark $\boldsymbol{C}$ to prove their distributorship to the recipient or copyright violation detection purposes. Next we explain the watermarking algorithm.

B. Embedding Algorithm

The encryption algorithm used is an additive privacy homomorphic one, so the watermark embedding is performed by using a robust additive watermarking technique. Since the embedding is done in the compressed ciphered bytestream, the embedding position plays a crucial role in deciding the watermarked image quality. Hence, for watermarking, we

consider the ciphered bytes from the less significant bit planes of the middle resolutions, because inserting watermark in ciphered bytes from most significant bit planes degrades the image quality to a greater extent. Also, the higher resolutions are vulnerable to transcoding operations and lower resolution contains a lot of information, whose modification leads to loss of quality. We in our experiments study the impact on quality of watermarking in this compressed-encrypted domain. We show how the watermark can be inserted in less significant bit planes of middle resolutions without affecting the image quality much. Since the embedding and detection are done on integer domain, the watermark is added after rounding off to the nearest integer for SCS-QIM and RDM. The rounding off process decreases the watermark power or in other words introduces noise and its effect on detection performance is given in the Appendix. We now explain the embedding process.

1) SS: Hartung *et al.* in [20] proposed a spread spectrum watermarking scheme. The embedding process is carried out by first generating the watermark signal $\boldsymbol{W}$ by using watermark information bits $\mathbf{b}$, chip rate r and PN sequence $\boldsymbol{P}$. The watermark information bits $\mathbf{b} = \{b_i\}$, where $b_i = \{1, -1\}$, are spread by r, which gives

$$a_j = b_i, \quad ir \leq j < (i+1)r. \tag{3}$$

The sequence a_j is then multiplied by $\alpha > 0$ and $\boldsymbol{P}$. The watermark signal $\boldsymbol{W} = \{w_j\}$, where

$$w_j = \alpha a_j p_j \tag{4}$$

where $p_j = \{1, -1\}$. The watermark signal generated in (4) is added to the encrypted signal $\boldsymbol{C}$, to give the watermarked signal $\boldsymbol{C_W}$

$$\boldsymbol{C}_W = \boldsymbol{C} + \boldsymbol{W} = c_{w_i} = c_i + w_i \quad \forall i = 0, 1, \ldots, L-1. \tag{5}$$

Here, $\boldsymbol{C}$ and $\boldsymbol{W}$ can be considered to be $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively, of (2). Although $\boldsymbol{W}$ is added in plaintext form, it can be considered to be encrypted using key $\boldsymbol{K}_2$ such that $\boldsymbol{K}_2$ is a stream of bytes with value zero, then $\boldsymbol{C}_2 = \boldsymbol{W}$. In other words, as $\boldsymbol{M}_2$ in (2) corresponds to $\boldsymbol{W}$ of (5), $\boldsymbol{M}_2$ can be assumed to be encrypted using a byte key stream $\boldsymbol{K}_2 = \{k_{2_i}\}\ \forall i = 0, 1, \ldots, L-1$. Now, if $\{k_{2_i}\} = 0\ \forall i = 0, 1, \ldots, L-1$, then the encrypted value of $\boldsymbol{M}_2$ denoted by $\boldsymbol{C}_2$ is

$$c_{2_i} = (m_{2_i} + k_{2_i}) \bmod 255\ \forall i = 0, 1, \ldots, L-1. \tag{6}$$

Thus, we get $\boldsymbol{C}_2 = \boldsymbol{M}_2$, i.e., encryption of $\boldsymbol{M}_2$ still produces $\boldsymbol{M}_2$ as addition of zero do not make any change in (6). Also, the decryption key $\boldsymbol{K} = \boldsymbol{K}_1 + \boldsymbol{K}_2$ for decrypting $\boldsymbol{C}_1 + \boldsymbol{C}_2$ can be written as $\boldsymbol{K} = \boldsymbol{K}_1$ as $\boldsymbol{K}_2 = \boldsymbol{0}$. Thus, according to homomorphic property we can write

$$\begin{aligned} &D(\boldsymbol{C}_1 + \boldsymbol{C}_2, \boldsymbol{K}(= \boldsymbol{K}_1 + \boldsymbol{K}_2)) \\ &\quad \equiv D(\boldsymbol{C}_1 + \boldsymbol{M}_2, \boldsymbol{K}(= \boldsymbol{K}_1)) \\ &D(\boldsymbol{C}_1 + \boldsymbol{M}_2, \boldsymbol{K}) = \boldsymbol{M}_1 + \boldsymbol{M}_2. \end{aligned} \tag{7}$$

If c_{w_i} is more than 255, a lesser strength (may be zero as well) of watermark is added such that c_{w_i} remains below 255. Thus decrypting C_W, we get $M + W$ since W is inserted in plain text form.

2) SCS-QIM: In [21], Eggers *et al.* proposed SCS scheme for watermark embedding. In this scheme, given a watermark strength, we choose a quantizer from an ensemble of quantizers to embed the watermark. For a binary watermark $w \in \{0, 1\}$, the quantizer can be chosen as

$$\boldsymbol{U} = (l + k_{qim_i})\beta\Delta + w\beta\Delta/2 \quad \forall i = 0, 1, \ldots, L-1. \tag{8}$$

w introduces a shift in the quantizer and l gives the different sets of quantizers. Also, for making the codebook secure a random sequence K_{qim} can be chosen. The embedding scheme is then

$$\begin{aligned} q_i = Q_\Delta &\left(c_i - \Delta\left(\frac{w_i}{2} + k_{qim_i}\right)\right) \\ &- \left(c_i - \Delta\left(\frac{w_i}{2} + k_{qim_i}\right)\right) \\ &\forall i = 0, 1, \ldots, L-1 \end{aligned} \tag{9}$$

where $Q_\Delta(\cdot)$ denotes scalar uniform quantization with step size Δ. The watermark sequence is then given by

$$\boldsymbol{W} = \beta \mathbf{q} \tag{10}$$

and the embedding is done as

$$\boldsymbol{C}_W = \boldsymbol{C} + \boldsymbol{W}. \tag{11}$$

Here also, like in Section II-B1, $\boldsymbol{C}$ and $\boldsymbol{W}$ can be considered to be $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively, of (2).

The power of quantization error is given as $\mathrm{Exp}(q^2) = \Delta^2/12$, considering a uniform distribution for q. So, for a given watermark power σ_w^2

$$\beta = \sqrt{\frac{\sigma_w^2}{\mathrm{Exp}(q^2)}} = \sqrt{\frac{12\sigma_w^2}{\Delta^2}}. \tag{12}$$

3) RDM: In [22], Gonzalez *et al.* proposed a watermarking scheme based on quantization of the ratio of host signal to a function $\boldsymbol{g}(\cdot)$ defined later in this section. The quantizers are given by

$$Q'_\Delta = 2\Delta + w\Delta/2 \tag{13}$$

where $w \in \{-1, 1\}$ is the watermark information to be embedded in the host element. The embedding rule is then

$$c_{w_i} = g(\boldsymbol{c}_{w_{i-1}}) Q'_\Delta(c_i / g\left(\boldsymbol{c}_{w_{i-1}}\right)) \quad \forall i = 1, \ldots, L-1 \tag{14}$$

where c_{w_i} and $c_{w_{i-1}}$ are the current and previous watermarked samples, and Q'_Δ is as given in (13). Notice that c_{w_i} is an amplitude enhanced version of scaled-quantized c_i. Thus we can write

$$w_i = c_{w_i} - c_i \tag{15}$$

which gives the additive nature of watermark. The function $\boldsymbol{g}(\cdot)$ is chosen such that the scheme is robust against amplitude scaling attacks and is given by

$$\boldsymbol{g}\left(\boldsymbol{c}_{w_{i-1}}\right) = \left(\frac{1}{L_m} \sum_{j=i-L_m}^{i-1} \left|c_{w_j}\right|^\gamma\right)^{1/\gamma}, \quad \gamma \geq 1. \tag{16}$$

One of the drawbacks with this scheme is that the watermarked sample may differ from the original sample to a large extent due to the function $\boldsymbol{g}(\cdot)$ used for quantization. So, we scale $\boldsymbol{g}(\cdot)$ by a constant factor S_c known at both encoder and decoder to control the amount of watermark added. The impact of scaling on the decoding of watermark is given in the Section III-E.

Thus, watermark embedding is carried out in compressed-encrypted domain, and the watermarked content is then distributed by the distributors. Also, it is shown in Section II-C that the watermarked quality can be controlled in a predictable manner. The security of the schemes SS, SCS-QIM and RDM is discussed in Section III-D.

C. Watermark Detection

The watermark can be detected either in encrypted or decrypted compressed domain. We also discuss the uncompressed domain detection in case of SS technique. We will first discuss the detection in encrypted domain followed by decrypted domain. The code-book in case of SCS-QIM and RDM is formed as described in Sections II-B2 and II-B3, respectively.

1) Encrypted Domain Detection: In encrypted domain, as shown in Fig. 1, $\boldsymbol{C_W}$ is directly given to the watermark extraction module and the detection process is as follows.

a) SS: The received encrypted-watermarked signal $\boldsymbol{C_W} = \boldsymbol{C}+\boldsymbol{W}$ is applied to the correlator detector. It is multiplied by PN sequence $\boldsymbol{P}$ used for embedding, followed by summation over chip-rate window r, yielding the correlation sum S_i. Assuming zero correlation between $\boldsymbol{C}$ and $\boldsymbol{P}$

$$S_i = \sum_r \left(c_{w_j} p_j\right) = \sum_r (c_j + w_j) p_j = b_i \sigma_p^2 \alpha r. \tag{17}$$

The first term in (17), i.e., $c_j p_j$, is zero if $\boldsymbol{C}$ and $\boldsymbol{P}$ are uncorrelated. However, this is not always the case for real compressed data. Thus, we can apply the non-blind detection technique, i.e., subtract away $\boldsymbol{C}$ from $\boldsymbol{C_W}$ to remove the correlation effect completely. Thus get a better watermark detection rate. The sign of S_i gives the watermark information bit:

$$\operatorname{sign}(S_i) = \operatorname{sign}\left(b_i \sigma_p^2 \alpha r\right) = \operatorname{sign}(b_i) = b_i. \tag{18}$$

However, the distributors can also use prefiltered (semi-blind) detection technique. Incase of ownership proving applications the prefiltered (semi-blind) detection technique may be required. In this case, the watermarked message is first passed through a high pass filter to reduce the cross-talk between watermark signal and host samples. The filtered message is then multiplied by PN sequence and thereby extracting the watermark.

b) SCS-QIM: Watermark is estimated by quantizing the received signal to the nearest data in the codebook, as given

$$\hat{w} = Q_\Delta\left(c_{w_i}\right) - c_{w_i} \quad \forall i = 0, 1, \ldots, L-1. \tag{19}$$

If $\hat{w}$ is close to zero, then watermark bit $\hat{w_e} = 0$ is extracted and if close to $\pm\Delta/2$, then bit $\hat{w_e} = 1$ is retrieved.

c) RDM: The detection of watermark is performed by the minimum distance criteria using the following equation, as given:

$$\hat{w} = \arg\min_{1,-1} \left(\frac{c_{w_i}}{g\left(\boldsymbol{c}_{w_{i-1}}\right)} - Q'_\Delta\left(\frac{c_{w_i}}{g\left(\boldsymbol{c}_{w_{i-1}}\right)}\right)\right)^2 \quad \forall i = 1, \ldots, L-1. \tag{20}$$

Here, Q'_Δ gives two quantizers belonging to bits 1 and -1. The distance is computed corresponding to both the quantizers and the one which gives minimum distance gives the watermark bit.

2) Decrypted Domain Detection: The received compressed-encrypted watermarked image is first passed through the decryption module, shown in Fig. 1, and is decrypted using (21), which defines the corresponding byte by byte decryption for the encryption defined in (1). The keystream $\boldsymbol{K}$ can be generated as given in Section III-A. The received signal $\boldsymbol{C_W}$ is decrypted to give $\boldsymbol{M_W}$ as

$$\begin{aligned} \boldsymbol{M_W} = \boldsymbol{D}(\boldsymbol{C_W}, \boldsymbol{K}) &= (c_{w_i} - k_i) \bmod 255 \\ &\qquad \forall i = 0, 1, \ldots, L-1 \\ &= (c_i + w_i - k_i) \bmod 255 \\ &= m_i + w_i \\ &= m_{w_i}. \end{aligned} \tag{21}$$

It can be seen from (21) that $m_{w_i} = m_i + w_i$, the watermarked compressed byte stream m_{w_i} is merely addition of compressed byte stream m_i, and the watermark signal w_i. Thus by controlling the strength of w_i, choice of resolution levels and bit planes, the quality of the watermarked signal could be easily controlled. The watermarked quality would be poor if we pick up more number of resolution levels and bit planes to watermark, but the watermark embedding capacity would be high and vice versa.

For SS detection, the embedded watermark information $\boldsymbol{W}$ can be estimated from $\boldsymbol{M_W}$ using correlation detector even without the knowledge of the corresponding originals $\boldsymbol{M}$ or $\boldsymbol{C}$. However, $\boldsymbol{M}$ and $\boldsymbol{P}$ may not always be uncorrelated and hence the noise due to M may not be completely eliminated. Therefore to obtain better detection results, we can encrypt $\boldsymbol{M_W}$ with $\boldsymbol{K}$ which gives $\boldsymbol{C_W}$ and removing $\boldsymbol{C}$ gives

$$S_i = \sum_r (w_j p_j) = \sum_r \alpha a_j p_j p_j = b_i \sigma_p^2 \alpha r. \tag{22}$$

Thus, the sign of S_i gives the watermark information bit

$$\operatorname{sign}(S_i) = \operatorname{sign}\left(b_i \sigma_p^2 \alpha r\right) = \operatorname{sign}(b_i) = b_i. \tag{23}$$

Although we have not considered the addition of noise by the adversary in (22), an elaborate analysis of the watermark detection performance under the addition of noise is discussed in Appendix A.

Similarly for SCS-QIM and RDM, the decrypted message $\boldsymbol{M_W}$ along with cipher key $\boldsymbol{K}$ is fed to the watermark extraction module. The signal $\boldsymbol{M_W}$ is encrypted with the key $\boldsymbol{K}$ with the method described in Section II-A. Thus, we get the ciphered-watermarked signal $\boldsymbol{C_W}$ and the watermark is detected using (19) and (20), respectively.

Although the detection in decrypted domain replicates the encrypted domain detection, this is possible due to homomorphism property of cryptosystem. The homomorphic property allows us to embed the watermark in a predictable way. The homomorphism property is discussed in Section II-A. Here, $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ can be treated as encrypted host signal and watermark signal, respectively. From (2), we see that the result is addition of plain host and watermark signal. In this way we can insert the watermark signal such that it is detectable as well as control the watermarked quality. In case the cryptosystem does not follow

homomorphism, the watermark information will become unpredictable once the content is decrypted thus rendering detection in decrypted domain difficult. The quality of the decompressed content will also be unpredictable in this case.

We also give an alternative watermark detection procedure where the encryption key $\boldsymbol{K}$ is not required by the distributor for the extraction of watermark. In this extraction process, when the distributor wants to extract the watermark from a suspected decrypted content, the distributor first sends the suspected decrypted content to the owner to re-encrypt. The owner then encrypts the suspected content using the key $\boldsymbol{K}$ and sends it to the distributor. The distributor then carries out the watermark extraction using the technique described in Section II-C. Thus, the distributor can extract the watermark embedded by him/her without having to know the encryption key $\boldsymbol{K}$. This simple protocol may increase the communication overhead between the owner and distributor by two messages per watermark extraction process and the computation overhead of the owner by one encryption operation. However, such watermark extraction would happen only when there is a suspected content to be verified, which would not happen very often. Thus this increase in communication and computation overheads is not significant.

3) Uncompressed Domain Detection: Let $I_{\mathrm{DW}}, I_{\mathrm{DU}}, I_{\mathrm{DWA}}$ denote decompressed-watermarked image, decompressed-original image, and decompressed-watermarked-attacked image. Then the watermark signal in decompressed domain can be computed as $\hat{W} = I_{\mathrm{DU}} - I_{\mathrm{DW}}$ and in case of attack, $\check{W} = I_{\mathrm{DU}} - I_{\mathrm{DWA}}$. For detection, a correlation measure between embedded and attacked watermark signal is computed as

$$\mathrm{corr}(\hat{W}_i, \check{W}) = \frac{E\left[\left(\hat{W}_i - \mu_{\hat{W}_i}\right)(\check{W} - \mu_{\check{W}})\right]}{\sigma_{\hat{W}_i}\sigma_{\check{W}}} \quad \forall i = 1, 2, \ldots, N_w \tag{24}$$

where $\mathrm{corr}(\cdot,\cdot)$ denotes the correlation measure, $E[\cdot]$ denotes the expectation operator, μ denotes the mean, and σ^2 denotes the variance. The correlation value against different watermarks is measured, i.e., $\mathrm{corr}(\hat{W}_i, \check{W})\ \forall i = 1, 2, \ldots, N_w$, where N_w denotes the number of watermarks, and the watermark $\hat{W}_i$ with maximum correlation value gives the embedded watermark.

III. Discussion

A. Key Stream Generation

The keystream is generated at the encryption and decryption site using RC4 cipher [23]. For encryption, a secret seed $\boldsymbol{S}$ is applied to RC4 cipher which in turn generates the keystream $\boldsymbol{K}$. In order to generate the same key $\boldsymbol{K}$ at the decryption site, the seed $\boldsymbol{S}$ must be delivered to the decryption site through a secret channel. Once the seed $\boldsymbol{S}$ is received, it can be applied to RC4 cipher to generate the key stream $\boldsymbol{K}$ which is further used for decryption.

B. Domain of Encryption

The compressed-encrypted domain is chosen for watermarking as the content is often distributed in this domain. In such a scenario, the domain of encryption plays an important role because of the trade-off between compression efficiency and security. Encryption is generally applied either in bitstream domain, i.e., the final compressed codestream or in the compression pipeline itself, i.e., spatial or transform steps. Different schemes have been evaluated in [24] for the encryption of JPEG2000 compressed images. Fig. 2 shows the quality versus compression rate plot when the encryption algorithm is applied in the spatial domain as well as without encryption. PSNR of the "plain data" in Fig. 2 is the same as the PSNR performance of the proposed encryption scheme (when measured between original and decrypted-decompressed images). Thus the quality of the image drops drastically with increase in compression rate and is unsuitable for practical applications. However, the encryption method given in Section III-A does not affect the quality of image and can be used for confidentiality and other purposes.

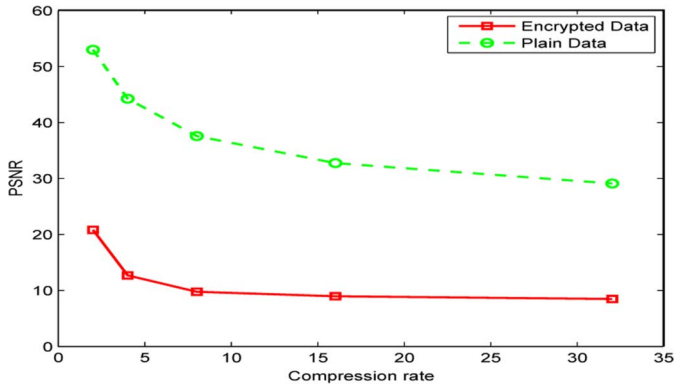


Fig. 2. PSNR degradation in spatial domain.

C. Security of Encryption Algorithm

The RC4 cryptosystem that we have used is a very well-known technique and is believed to be secure after first few hundred bytes are discarded [25], [26]. Henceforth, we will assume that first few hundred bytes are discarded. Now we establish the security of the applied encryption scheme based on Shannon's theory of security [27]. We assume that $\boldsymbol{M} \sim U[0, 255]$ and let $\boldsymbol{P_M}$ and $\boldsymbol{P_{K'}}$ denote the probabilities of occurrence of random variables $\boldsymbol{M}$ and $\boldsymbol{K'}$, respectively, so the amount of information contained in $\boldsymbol{M}$ is given by

$$\boldsymbol{H_M} = -\sum \boldsymbol{P_M} \log \boldsymbol{P_M}. \tag{25}$$

Similarly, computing for a truly random key $\boldsymbol{K'} \in [0, 255]$

$$\boldsymbol{H_{K'}} = -\sum \boldsymbol{P_{K'}} \log \boldsymbol{P_{K'}}. \tag{26}$$

For perfect secret systems the amount of information $\boldsymbol{H_M}$ can be hidden completely if $\boldsymbol{H_{K'}} >= \boldsymbol{H_M}$ [27], which is correct if the key is truly random. However, the RC4 keystream $\boldsymbol{K}$ is a pseudo-random sequence having a different distribution than that of a random keystream $\boldsymbol{K'}$. Having said that, we will now show that under certain bounds the RC4 keystream can be assumed to behave like a truly random sequence, thereby proving the security.

In [28] Fluhrer *et al.* showed that $\approx 2^{30.6}$ bytes are sufficient to discriminate RC4 output cipher from a truly random sequence. However, this bound is much higher than the size of compressed image which is of the order of few kilo bytes. Thus, the size of

the compressed data is not sufficient to clearly distinguish between RC4 cipher and a truly random stream and hence, we assume within the limits of aforementioned bounds that the RC4 keystream can be approximately modeled as a truly random sequence, which establishes the security of our system. Another attack on a particular mode of operation of RC4 occurs when the same secret seed $\boldsymbol{S}$ is used multiple times. In this case, the seed $\boldsymbol{S}$ is sent as a concatenation of $\boldsymbol{S}$ with a key (referred to as Initialization Vector IV, need not be secret). In this case, the attacker tries to find the secret seed $\boldsymbol{S}$ by observing the output streams for different IV values and a fixed $\boldsymbol{S}$, and can recover the keystream $\boldsymbol{K}$ without much time complexity [29]. However, this attack can be overcome by using different secret keys, in which case the attack scenario is same as [28]. Other encryption schemes such as SNOW 3G can also be used to overcome this attack [30].

In [31] Mantin *et al.* proposed an attack by observing first two output words of the cipher. This information is used to deduce partial information of plaintext by analyzing different ciphertexts produced from the same plaintext using different secret keys. However, this attack does not recover the key. Moreover, it becomes insignificant for output word size of more than 5 bits.

D. Security of Watermarking Algorithm

The watermarking algorithm is as robust as underlying watermarking schemes, i.e., SS, SCS-QIM, and RDM. The attacks can be performed either in encrypted or decrypted compressed domain to retrieve or destroy the watermark. The attacks are considered in compressed domain since watermark detection for ownership verification, traitor tracing, or copyright violation detection can easily be done as the content is often copied and distributed in compressed format. The watermark detection in case of uncompressed domain is also discussed in Section II-C3. The watermark detection performance against attacks like additive Gaussian noise, filtering, and amplitude scaling is given in Section IV. The robustness, for SS scheme, against filtering, such as 1×5 median filter and scaling attack can be improved by increasing the chip rate and estimating the scale factor [21], respectively. However in case of mean and Gaussian filtering, when the watermarked samples are replaced by the prediction made from the neighboring samples, the replaced samples may be very different compared to the unfiltered watermarked samples. This is due to the fact that the watermarked samples are uncorrelated and the prediction from the neighboring samples may not be a good approximation of the unfiltered watermarked samples. Thus, it leads to the addition of huge amount of noise to the watermarked samples which may be difficult to remove and the detection performance is not effective. Further, SCS-QIM and RDM are not robust against filtering attack. Collusion attacks can be made ineffective by using collusion resistant codes to identify all or at least groups of users involved in collusion [32].

In some cases, attacker tries to confuse ownership by creating a fake original or fake watermarked data. In this case, a watermark signal dependent on hash of the original or watermarked content and a private identifier (known only to the watermark embedding party) can be used for embedding [33]. Thus the party which embeds its watermark first in the content can provide the hash of the original content and identifier to detect

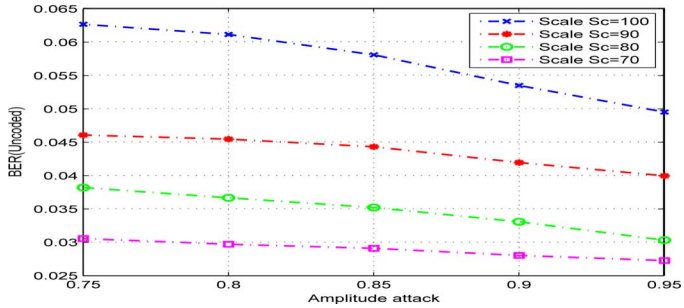


Fig. 3. BER versus amplitude scaling RDM.

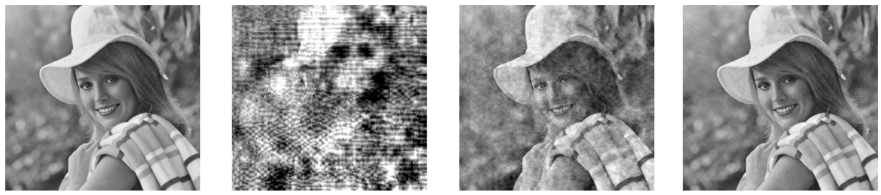
his own watermark. However, the party which embeds subsequently has the watermarked content but cannot provide hash of original content and identifier and cannot detect his own watermark in the original content.

As in any traitor tracing watermarking techniques, here also exists false implication concerns for distributors, i.e., it is possible that a distributor can be falsely implicated by the owner. The owner may estimate the distributor's watermark and then use it for defaming the distributor by embedding the distributor's watermark in other illegal contents. This problem can easily be tackled using Memon *et al.*'s buyer-seller watermarking protocol [34], which can easily be integrated into our proposed compressed encrypted domain watermarking technique. In buyer-seller watermarking protocol a watermark is jointly created by the buyer and seller. In order to jointly create the watermark, the buyer sends a valid watermark in an encrypted form to the owner. Also, the encryption is commutative with respect to permutation. The seller applies a random permutation to the encrypted watermark which is not known to buyer. The jointly created watermark is then embedded in the content in the encrypted domain. Thus neither the seller (owner) nor the buyer (distributor) gets to know this jointly created watermark, which is embedded in encrypted domain. Thus, owner's watermark, distributor's watermark and this jointly created watermark would be in the watermarked media. These three watermarks would provide the necessary mechanisms for traitor tracing, copyright management and distributor protection against false implication.

E. Effect of Scaling in RDM Detection

In case of watermarking using RDM, the amount of watermark power embedded varies to a great extent due to varying quantization step size. Towards this, Abrardo *et al.* proposed a watermarking scheme using trellis coded quantization [35]. However, it still uses the function $\boldsymbol{g}(\cdot)$ which might not improve the watermarked quality when $\boldsymbol{g}(\cdot)$ itself varies highly. To overcome this drawback, we scale the step size or the function $\boldsymbol{g}(\cdot)$, to suppress this high variation. Also, the watermark power can be controlled using the desired scale. However, we are interested in dealing with the impact of scaling on detection performance. The quantization of signal c_i with the quantizer Q'_Δ can be written as $Q'_\Delta(c_i, \Delta)$. When message c_i is scaled with a constant S_c, then we have

$$Q'_\Delta(c_i, \Delta/S_c) = \frac{1}{S_c} Q'_\Delta(S_c c_i, \Delta). \tag{27}$$



(a) (b) (c) (d)

Fig. 4. (a) Original image. (b)–(d) Watermarked images for bit planes $l = 10, 8$, and 6, respectively.

Thus the properties of both the quantizers change equivalently and as S_c is known, it does not impact the detection performance. Figs. 3 and 12 shows the detection error rate against noise and amplitude scaling attack. It can be seen from Fig. 3 that the BER does not change significantly for a given scale. Although the BER changes for a given scale, it is caused because of the noise power added due to the processing in integral domain. However, when the scale is decreased, the watermark power increases which attributes to decrease in BER.

IV. Experimental Results

Experiments have been carried out for 1000 grayscale images (of size 512×512 [36], and 512×384 [37]) over six different compression resolutions (0–5) and at a compression rate of 4. The value of α is set between 0 to 3. Convolutional codes with a Viterbi decoding decision are used as error correction codes (ECC) to improve the detection performance only for SCS-QIM and RDM due to the impact in error rate caused by round-off process.

A. Quality versus Payload Trade-Off

Fig. 4 gives the original image and watermarked images for $l = 10, 8$, and 6. In Fig. 4, the payload capacity for the number of watermarked bitplanes $=l$ means the bitplanes $l, l-1, \ldots, 1$ are watermarked. For $l = 10$ means all the bitplanes except the most significant bitplane are watermarked. Similarly, $l = 1$ means only the least significant bitplane is watermarked. Table I gives the average of PSNR and payload for different l when all as well as middle resolutions 1, 2, and 3 are watermarked. It is clear from Fig. 4 and Table I that when middle resolutions are watermarked, embedding a watermark for $l \geq 8$ is undesirable as the quality is poor. While when all resolutions are watermarked $l = 6$ can be used for embedding. From Table I we can infer that when resolutions 1, 2, and 3 are watermarked, $l = 7$ or $l = 6$ gives good trade-off between payload and watermarked image quality. Thus $l = 7$ can be used for watermarking if quality can be compromised while $l = 6$ is suitable for applications which demand high quality. On the other hand, if all the resolutions are considered for watermarking, then $l = 6$ should be used.

1) Embedding Capacity: The average payload capacity versus number of bit planes(l) watermarked under different resolutions using SCS-QIM scheme is given in Fig. 5. As the payload capacity and PSNR does not vary too much for the three schemes, only the average payload and average PSNR for all the resolutions are plotted for SS and RDM schemes, while the details are provided in Tables I and II. The average size of the compressed images is 48 kilo bytes. Average payload capacity is given here as the ratio of the average embedded

TABLE I
PSNR and Payload for Different l for Image Size 512×384

l	All resolutions		Resolutions 1, 2 and 3	
	PSNR(dB)	Payload(bits)	PSNR(dB)	Payload(bits)
11	7.7	44877	11.7	7232
10	11.4	42534	16.54	7135
9	16.81	38663	20	6767
8	21.06	30412	24.6	6003
7	27.45	19489	30.1	4894
6	33.6	9807	35.55	3551

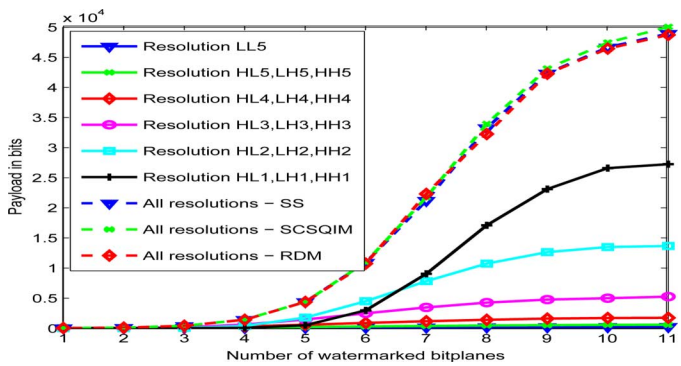


Fig. 5. Average payload capacity versus number of watermarked bit planes (l) under different resolutions.

number of bits to the average compressed stream size (in bytes), where average is computed as a simple mean. In Fig. 5 it is clear that as we move from resolutions LL5 (lowest) to HH1, HL1, and LH1 (highest) the payload capacity increases, and the quality of the image is good. The increase in payload capacity is due to increase in size of dimensions of higher resolutions, generating more number of compressed bytes, which provides more space for embedding. In Fig. 5, it can be seen that a lower resolution, say 4 (HL2, LH2, HH2) gives more capacity than a higher resolution, say 5 (HL1, LH1, HH1) at $l = 6$, due to the fact that rate-distortion optimization is followed in JPEG2000, i.e., truncation of codestream according to a given bit rate which achieves minimum distortion and distortion caused by truncating codestream from a higher resolution (5) is less than that of a lower resolution (4).

2) Watermarked Image Quality: Fig. 6 shows PSNR against l for the payload given in Fig. 5, where PSNR is calculated as

$$\mathbf{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - I_w(i,j))^2 \tag{28}$$

$$\mathbf{PSNR} = 10 \log_{10}(255^2/\mathbf{MSE}). \tag{29}$$

Mean square error (MSE) is the sum of squares of difference between I and I_w. In Fig. 6, for resolution 4 (HL2, LH2,

TABLE II
BER, PAYLOAD, AND PSNR FOR SS, SCS-QIM, AND RDM (IMAGE SIZE 512 × 384)

Scheme	Attack		All resolutions (l = 6)		Resolutions 1, 2 and 3 (l = 7)		Coderate
	Noise	Scale	Payload(bits)	PSNR(dB)	Payload(bits)	PSNR(dB)	
SS (non-blind)	0	0	9767	33.45	4677	30.15	1/32
SCS-QIM	10^{-4}	.5	9807	33.4	4894	30.1	1/6
RDM	10^{-3}	10^{-2}	9598	33.85	4126	31.22	1/10

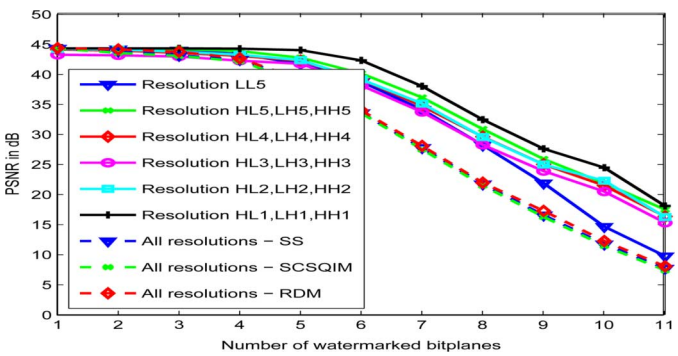


Fig. 6. Average PSNR versus number of watermarked bit planes (l) under different resolutions.

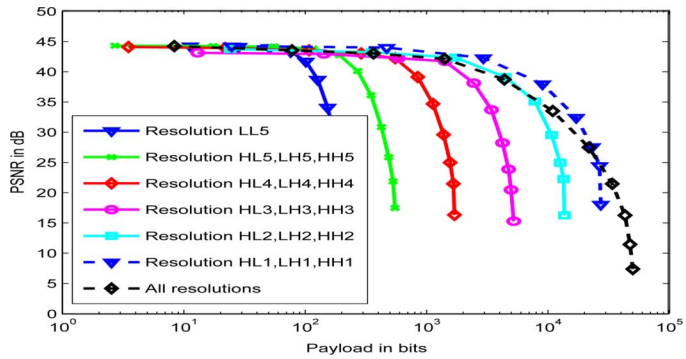


Fig. 7. Average payload versus PSNR under different resolutions.

HH2) and $l = 6$, the PSNR is lesser than resolution 1 (HL5, LH5, HH5) as the embedding capacity for resolution 4 is much higher compared to that of resolution 2 which can be seen from Fig. 5. However, the resolution 5 (HL1, LH1, HH1) has a higher PSNR in spite of high embedding capacity because the degradation caused due to watermarking higher resolution is lesser than caused by watermarking a lower resolution. Fig. 7 gives the payload capacity and watermarked image quality tradeoff. From Fig. 7, we can infer that the capacity drops sharply only when higher significant bitplanes are watermarked. Also the drop in quality and capacity increases as we proceed from higher resolutions towards lower resolutions.

The degradation of image quality due to watermarking lower resolutions especially LL5 is more than that caused by watermarking higher resolutions. This effect is more prominent for $l > 6$ as seen in Fig. 6. This is because the higher resolutions does not carry much of the relevant information, modifying which degrades the image quality to a lesser extent. However, the codestream from higher resolutions like 4 and 5 might be truncated more than the codestream from middle resolutions 1, 2, and 3 to meet the bit rate or bandwidth requirements because the distortion in the latter case will be more. Thus the bitplanes $l = 7$ of middle resolutions provide a good region for watermarking. The payload capacity for Lena, $l = 7$, considering all the resolutions, is 29 748 bits with an average PSNR of 32.58 dB, whereas when resolutions 1, 2, and 3 are considered the capacity is 7024 bits with an average PSNR of 35.22 dB, for a compressed image of size 64 kB. Figs. 8 and 9 show the original image, unwatermarked-decompressed image, encrypted image and watermarked-decompressed image for resolutions 1, 2, and 3 as well as all resolutions.

B. Performance Against Attacks

1) Encrypted/Decrypted Domain Performance:

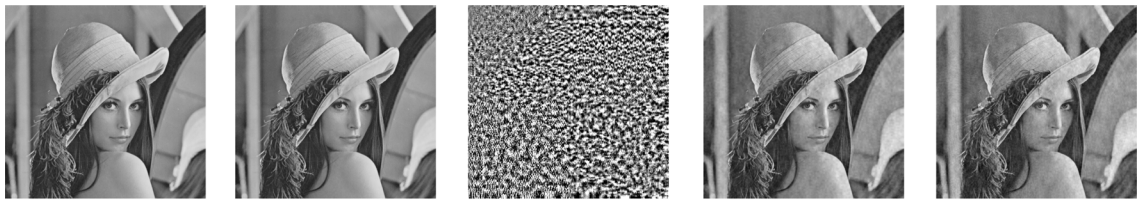
a) SS (Non-Blind Detection): The performance of the SS watermarking scheme against noise is given in Fig. 10 for non-blind detection. The bit error rate (BER) is defined as the ratio of number of incorrect watermark information bits retrieved to the total number of watermark information bits embedded and depends on the amount of noise added as well as chiprate. Also, the noise is added only in the selected bitplanes and resolutions used for watermarking. From Fig. 10, it is clear that the scheme gives good results even at lower watermark to noise ratio (WNR). Therefore for scenarios where WNR can be very low, the SS scheme can be used when the host signal is known at the decoder site. Also, BER for median filtering is 3.38×10^{-3} for a chip rate of 200 and the watermark can be detected without any bit error rate for scaling attack as the scale factor is estimated accurately. However, in case of blind detection, the watermark could not be detected, primarily because the noise due to host samples could not be eliminated due to presence of correlation between PN sequence and host samples. The BER is ≈ 0.5 in case of blind detection.

b) SCS-QIM (Blind Detection): The BER versus AWGN plot for SCS is given in Fig. 11 for different β. BER depends on the amount of noise added as well as coderate used for ECC. The coderate used is 1/6. Also, large values of Δ or β further decrease the BER, although it comes at the expense of increasing the watermark power. The round-off process leads to loss of performance as compared to original counterparts which is evident from Fig. 11. The BER also depends on the distribution of watermark power lost in round-off process and becomes significant at lower WNRs which is why the curves cross-over each other. WNR is given by

$$\mathbf{WNR} = 10 \log \left(\sigma_w^2 / \sigma_n^2 \right). \tag{30}$$

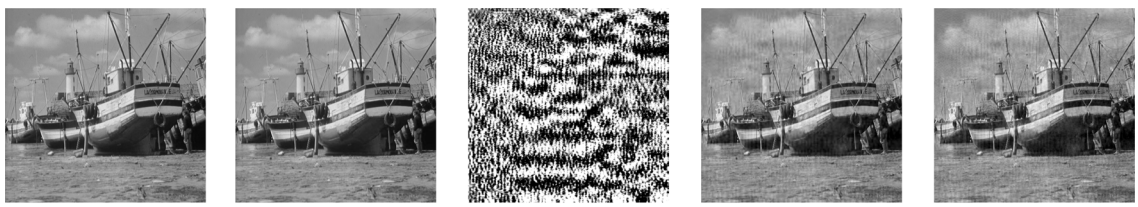
However, at higher WNRs, the watermark power dominates against this lost power, and the BER decreases.

c) RDM (Blind Detection): The BER versus AWGN plot for RDM is given in Fig. 12 for different S_c. The coderate used is 1/10. From Fig. 12, it is evident that the scheme performs well under high WNR conditions. RDM is highly robust against amplitude scaling attacks and gives an uncoded BER $\approx 10^{-2}$.



(a) (b) (c) (d) (e)

Fig. 8. (a) Original image. (b) Unwatermarked-decompressed image (47.8 dB). (c) Encrypted image. (d) Watermarked image (resolutions 1, 2, and 3) (35.22 dB). (e) Watermarked image (all resolutions) (32.58 dB).



(a) (b) (c) (d) (e)

Fig. 9. (a) Original image. (b) Unwatermarked-decompressed image (45.08 dB). (c) Encrypted image. (d) Watermarked image (resolutions 1, 2, and 3) (32.80 dB). (e) Watermarked image (all resolutions) (31.15 dB).

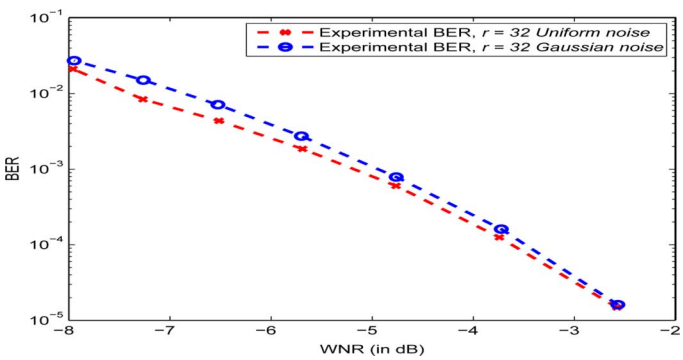


Fig. 10. BER versus WNR with $r = 32$ for SS watermarking scheme (non-blind detection).

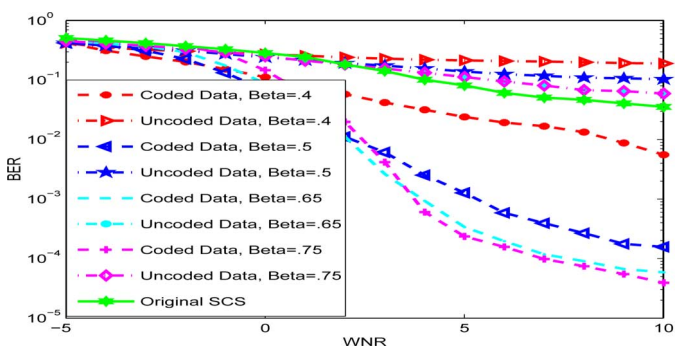


Fig. 11. BER versus WNR for SCS-QIM watermarking scheme for coded and un-coded watermark information (blind detection).

For a given WNR, the performance of SCS is better than RDM method against AWGN, which is clear from Figs. 11 and 12. Table II gives a comparison of BER and Payload for the three schemes at WNR $= 10$ dB. From Table II, we see that, for a given WNR, SCS-QIM provides better robustness against RDM with a higher watermark information carrying capacity. However, RDM gives better robustness against amplitude scaling

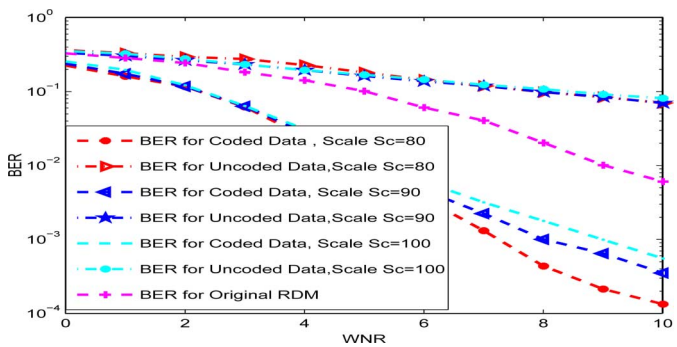


Fig. 12. BER versus WNR for RDM watermarking scheme with $\gamma = 1$ for coded and un-coded watermark information (blind detection).

attacks (uncoded BER $\approx 10^{-2}$) as compared to SCS-QIM for which even an amplitude factor of .95 gives BER (> 0.5). Although we have used codes with rate 1/6 and 1/10 for SCS-QIM and RDM, respectively, for experimental purpose, equal code rates will also lead to the same inference. Further, in Table II, we see that when $l = 6$, in case of all resolutions, due to the large size of higher resolutions (resolutions 4 and 5) compared to middle resolutions, the payload is higher than that of middle resolutions even when lesser number of bit planes are watermarked. Also, as lesser number of lower significant bit planes are used for embedding in case of all resolutions, the watermarked quality is better than the case when middle resolutions are watermarked.

2) Decompressed Domain Performance (Non-Blind Detection): In case of decompressed domain detection, under a noise attack of variance $\sigma_n^2 < \cdot 003$, the error in detection is $\approx 10^{-3}$. Fig. 13 gives the original, watermarked, and attacked images.

In case the decompressed attacked image is once again compressed (with a JPEG recompression with $QF > 70$) and distributed, the error in detection is $\approx 10^{-3}$. In order to do the watermark detection the recompressed image is first decompressed and then the watermark detection algorithm is applied.

TABLE III
COMPARISON OF SS, SCS-QIM, AND RDM

Scheme	Compressed domain			Decompressed domain		Watermark coderate
	Detection	Robustness		Detection	Robustness (Noise)	
		Noise	Scale			
SS	Non-blind	✓	✓	Non-blind	✓	Low
SCS-QIM	Blind	✓	×	Non-blind	✓	Moderate
RDM	Blind	✓	✓	Non-blind	✓	Moderate

(a) (b) (c)

Fig. 13. (a) Original image. (b) Watermarked-decompressed image (35.6 dB). (c) Attacked image (24.85 dB) $\sigma_n^2 = 0.003$.

In Table III, we compare the three watermarking techniques—SS, SCS-QIM, and RDM—based on the detection method, robustness in compressed/decompressed domain, and watermark coderate. From Table III, we can see that, in the compressed domain, the detection method applied for SS technique is non-blind while blind detection is applied for SCS-QIM and RDM. However, in decompressed domain only the non-blind detection is applied for all the three techniques. In terms of robustness in compressed domain, both the SS and RDM perform well against noise and scale attacks as compared to SCS-QIM which is robust against noise attack only. From Table II, we see that although RDM is not as robust as SCS-QIM against additive noise attack, it performs better against amplitude scaling attacks. Whereas, if the original content is available during detection, SS technique proves much better detection against noise and amplitude scaling attacks than SCS-QIM and RDM. In decompressed domain, all the techniques perform well against noise attack. Further, from Table III, we see that the watermark coderate is low for SS technique while it is moderate for SCS-QIM and RDM. However, SCS-QIM gives a better watermark information carrying capacity than SS and RDM, as can be seen in Fig. 5 and Table II.

V. CONCLUSION

In this paper we propose a novel technique to embed a robust watermark in the JPEG2000 compressed encrypted images using three different existing watermarking schemes. The algorithm is simple to implement as it is directly performed in the compressed-encrypted domain, i.e., it does not require decrypting or partial decompression of the content. Our scheme also preserves the confidentiality of content as the embedding is done on encrypted data. The homomorphic property of the cryptosystem are exploited, which allows us to detect the watermark after decryption and control the image quality as well. The detection is carried out in compressed or decompressed domain. In case of decompressed domain, the non-blind detection is used. We analyze the relation between payload capacity and quality of the image (in terms of PSNR and SSIM) for different resolutions. Experimental results show that the higher resolutions carry higher payload capacity without affecting the quality much, whereas the middle resolutions carry lesser capacity and the degradation in quality is more than caused by watermarking higher resolutions. However, higher resolutions might be truncated to meet the bandwidth requirements and in that case middle resolutions provide a good space for embedding. Further, as the watermark is embedded in different number of bitplanes l, in case of $l \geq 7$ the watermarked image quality may not be good. Therefore, $l < 7$ should be used for watermarking. We also study BER which shows that SS and SCS-QIM perform better under AWGN attack, whereas, RDM provides robustness against amplitude scaling attacks. ECC are also used to improve the detection rate. We also used a scaling factor to control the watermark power in RDM method and studied its impact in detection performance. The distortion due to the round-off process also plays a significant role in determining BER and the effect is also analyzed by comparing against the results of original watermarking schemes.

Future work aims at extending the proposed scheme to other image compression schemes such as JPEG, JPEG-LS. The major challenges with this compression schemes would be the lack of error resilience of the variable length codes used for encoding, and maintaining the compressed file size after encryption and watermarking so that the impact on compression gain is minimal.

APPENDIX A

We derive the error probabilities for the given watermarking schemes in the presence of noise. Let the noise be added by the adversary after decrypting the message. In order to simplify the mathematical analysis of the error probabilities we assume that the noise added by the adversary is uniformly distributed, then the modified message can be written as $\boldsymbol{M_W} + \boldsymbol{N}$. Now encrypting $\boldsymbol{M_W} + N$ with key $\boldsymbol{K}$ gives

$$\boldsymbol{Z} = (m_j + w_j + n_j + k_j) \bmod 255. \tag{31}$$

$\boldsymbol{Z}$ can be represented as

$$Z = \begin{cases} \boldsymbol{C_W} + \boldsymbol{N} \quad \forall 0 \leq \boldsymbol{C_W} + \boldsymbol{N} < 255 & \text{Case I} \\ (\boldsymbol{C_W} + \boldsymbol{N}) \bmod 255 & \\ \forall \{\boldsymbol{C_W} + \boldsymbol{N} \geq 255\} \bigcup \{\boldsymbol{C_W} + \boldsymbol{N} < 0\} & \text{Case II.} \end{cases} \tag{32}$$

For Case I, $\boldsymbol{Z} = \boldsymbol{C_W} + \boldsymbol{N}\ \forall 0 \leq \boldsymbol{C_W} + \boldsymbol{N} < 255$, the noise added by the adversary is N. But for Case II, $Z = (\boldsymbol{C_W} + \boldsymbol{N})$ mod 255 $\forall \{\boldsymbol{C_W} + \boldsymbol{N} \geq 255\} \bigcup \{\boldsymbol{C_W} + \boldsymbol{N} < 0\}$, and we can write $Z = \boldsymbol{C_W} + \boldsymbol{N} \pm 255 = \boldsymbol{C_W} + \boldsymbol{N'}$. Hence for Case 2, we can say that the noise takes the form $N' = N \pm 255$. We now compute the BER during the extraction of watermark from Z.

Let P_r be the probability of occurrence of Case I and hence the noise $\boldsymbol{N}$, then $1 - P_r$ would be the probability of occurrence

of Case II and hence the noise $\boldsymbol{N}'$. Then the resultant noise $\boldsymbol{N}_{\rm res}$ that contributes to the BER is coming from the noise $\boldsymbol{N}$ with probability P_r and $\boldsymbol{N}'$ with probability $1-P_r$. The resultant noise $\boldsymbol{N}_{\rm res}$ can be modeled as $\boldsymbol{N}_{\rm res}=\boldsymbol{N}\bigcup\boldsymbol{N}'$. We first compute the probability P_r and then derive the expression for BER.

Let us compute P_r. Here, $\boldsymbol{C_W}$ can be assumed to be uniformly distributed, i.e., $C_W \sim U[0,254]$ and $f_{\boldsymbol{C_W}}(c_w)=\sum_{i=0}^{254}\delta[c_w-i]/255$, where $c_w\in\mathbb{Z}$. Let the noise added by the adversary be uniformly distributed as $\boldsymbol{N}\sim U[-\tau,\tau]$. Then, the pdf of $\boldsymbol{N}$ is $f_{\boldsymbol{N}}(n)=\sum_{i=-\tau}^{\tau}\delta[n-i]/(2\tau+1)$, where $n\in\mathbb{Z}$. Since $\boldsymbol{C_W}$ and N are independent, the resultant pdf of the addition of both the variables is the convolution of individual pdfs [38]. Therefore, the resultant pdf $z=c_w+n$ is

$$f_{\boldsymbol{Z}}(z)=\sum_{n=-\infty}^{\infty}f_{\boldsymbol{C_W}}(z-n)f_N(n). \tag{33}$$

Since $f_{\boldsymbol{N}}(n)=\sum_{i=-\tau}^{\tau}\delta[n-i]/(2\tau+1)$ for $n\in[-\tau,\tau]$, we have

$$f_{\boldsymbol{Z}}(z)=\sum_{n=-\tau}^{\tau}\frac{f_{\boldsymbol{C_W}}(z-n)}{2\tau+1}. \tag{34}$$

Now $f_{\boldsymbol{C_W}}(z-n)=\sum_{i=0}^{254}\delta[z-n-i]/255$ when $0\le z-n<255$ and 0 elsewhere; then

$$\begin{aligned}f_{\boldsymbol{Z}}(z)&=\frac{1}{(2\tau+1)}\sum_{n=-\tau}^{z}f_{\boldsymbol{C_W}}(z-n)\quad\forall-\tau\le z<\tau\\&=\frac{1}{(2\tau+1)}\sum_{n=-\tau}^{\tau}f_{\boldsymbol{C_W}}(z-n)\quad\forall\tau\le z\le 254-\tau\\&=\frac{1}{(2\tau+1)}\sum_{n=z-254}^{\tau}f_{\boldsymbol{C_W}}(z-n)\\&\qquad\forall 254-\tau<z\le 254+\tau.\end{aligned} \tag{35}$$

Substituting $f_{\boldsymbol{C_W}}(z-n)$ in (35), we get

$$\begin{aligned}f_{\boldsymbol{Z}}(z)&=\frac{1}{(2\tau+1)}\sum_{n=-\tau}^{z}\sum_{i=0}^{254}\frac{\delta[z-n-i]}{255}\\&\qquad\forall-\tau\le z<\tau\\&=\frac{1}{(2\tau+1)}\sum_{n=-\tau}^{\tau}\sum_{i=0}^{254}\frac{\delta[z-n-i]}{255}\\&\qquad\forall\tau\le z\le 254-\tau\\&=\frac{1}{(2\tau+1)}\sum_{n=z-254}^{\tau}\sum_{i=0}^{254}\frac{\delta[z-n-i]}{255}\\&\qquad\forall\,254-\tau<z\le 254+\tau.\end{aligned} \tag{36}$$

Thus

$$\begin{aligned}P_r&=\frac{1}{2\tau+1}\sum_{n=-\tau}^{z}\sum_{i=0}^{254}\frac{\delta[z-n-i]}{255}(H[z]-H[z-\tau])\\&\quad+\frac{1}{(2\tau+1)}\sum_{n=-\tau}^{\tau}\sum_{i=0}^{254}\frac{\delta[z-n-i]}{255}\\&\quad\times(H[z-\tau]-H[z-(255-\tau)])\\&\quad+\frac{1}{(2\tau+1)}\sum_{n=z-254}^{\tau}\sum_{i=0}^{254}\frac{\delta[z-n-i]}{255}\\&\quad\times(H[z-(255-\tau)]-H[z-(255+\tau)])\end{aligned} \tag{37}$$

where $H[\cdot]$ is a heaviside step function.

We now derive the expression for BER which occurs while extracting the watermark from $\boldsymbol{Z}$. Since we have modeled the noise as $\boldsymbol{N}_{\rm res}=\boldsymbol{N}\bigcup\boldsymbol{N}'$, we can write $\boldsymbol{Z}$ as $\boldsymbol{Z}=\boldsymbol{C_W}+\boldsymbol{N}_{\rm res}$. Then

$$\begin{aligned}S_i&=\sum_r\left(c_{w_j}+n_{{\rm res}_j}\right)p_j\\&=\sum_r\left(c_j+w_j+n_{{\rm res}_j}\right)p_j.\end{aligned} \tag{38}$$

Since $\boldsymbol{C}=\{c_j\}\ \forall j$ is known at the decoder, then subtracting $\boldsymbol{C}$ from (38), we get

$$S_i=\sum_r\left(w_j+n_{{\rm res}_j}\right)p_j=\sum_r\alpha a_jp_j^2+\sum_r n_{{\rm res}_j}p_j. \tag{39}$$

Therefore, an error occurs when $\sum_r n_{{\rm res}_j}p_j<-\sum_r\alpha a_jp_j^2$ and $a_j=1$ or $\sum_r n_{{\rm res}_j}p_j>\sum_r\alpha a_jp_j^2$ and $a_j=-1$. Here the PN sequence $\boldsymbol{P}$ is having $\mu_p=0$ and variance σ_p^2, and the noise $\boldsymbol{N}_{\rm res}$ is also having zero mean and variance $\sigma^2_{n_{\rm res}}$. Therefore, the product $\boldsymbol{N}_{\rm res}\boldsymbol{P}$ has mean $\mu_{n_{\rm res}p}=0$ and variance $\sigma^2_{n_{\rm res}p}=\sigma^2_{n_{\rm res}}\sigma_p^2$. In the sum $V=\sum_r\boldsymbol{N}_{\rm res}P$, applying central limit theorem [20], [38], the pdf of the sum approaches a Gaussian distribution with mean $\mu_v=r\mu_{n_{\rm res}p}=0$ and variance $\sigma_v^2=r\sigma^2_{n_{\rm res}}\sigma_p^2$, where

$$\begin{aligned}\sigma^2_{n_{\rm res}}&=P_r\sigma_n^2+(1-P_r)\sigma^2_{n'}\\&=P_r\sigma_n^2+(1-P_r)\left(\sigma_n^2+255^2\right)\end{aligned} \tag{40}$$

where σ_n^2 and $\sigma^2_{n'}$ are the variances of $\boldsymbol{N}$ and $\boldsymbol{N}'$, respectively. Although a closed loop expression for a discrete Gaussian may be practically difficult to find, we can find the BER using a continuous Gaussian distribution with same mean and variance, as the BER is nothing but the summation of the probabilities $\sum_r n_{{\rm res}_j}p_j>r\alpha\sigma_p^2$.

The probability of error is given by [20]

$$\begin{aligned}&{\rm BER}\left(\sum_r n_{{\rm res}_j}p_j>r\alpha\sigma_p^2\right)\\&\qquad=\frac{1}{\sqrt{2\pi}\sigma_v}\int_{r\alpha\sigma_p^2}^{\inf}\exp^{-v^2/2\sigma_v^2}\,dv\end{aligned} \tag{41}$$

which can be written as

$${\rm BER}=\frac{1}{2}{\rm erfc}\left[(\sigma_p\sqrt{r}\alpha)/\left(\sqrt{2}\sigma_{n_{\rm res}}\right)\right]. \tag{42}$$

We now compare the analytical results with the experimentally obtained results. Fig. 14(a) gives the comparison for $r=200$, $\alpha=3$, and $\sigma_p=1$ under varying σ_n^2. From Fig. 14(a), we see that the BER for the analytical and experimental cases are similar, and the experimental BER is higher. The difference between the curves could occur due to many reasons such as the assumption of uniform distribution for the samples in a given chiprate interval may not hold valid. Also, the noise may not be completely independent to the pseudo random sequence $\boldsymbol{P}$.

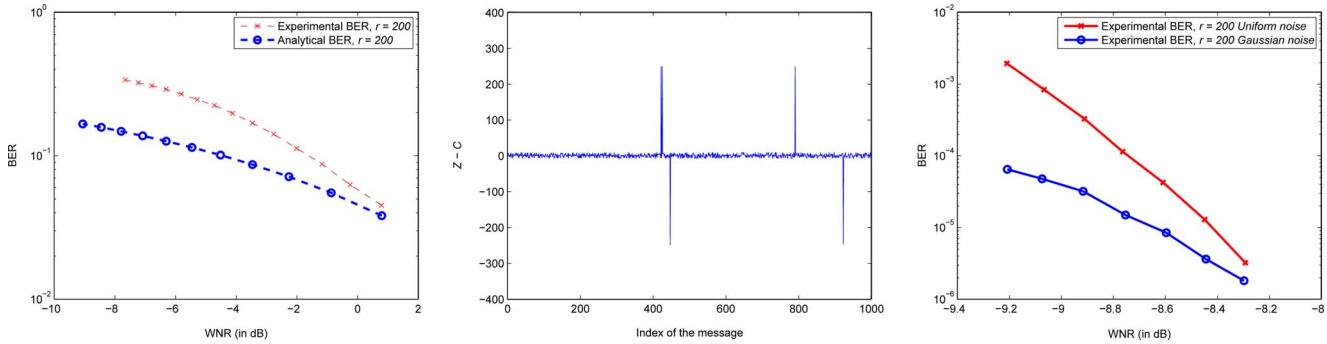


Fig. 14. (a) BER analytical versus experimental. (b) Values of $Z - C$ versus index of the message. (c) BER after removal of the peaks.

In addition, the experimentally obtained results can be improved further by identifying the peaks which occur due to the case $\{\boldsymbol{C_W} + \boldsymbol{N} < 0\} \bigcup \{\boldsymbol{C_W} + \boldsymbol{N} \geq 255\}$ while extracting the watermark from $\boldsymbol{Z}$ after performing $\boldsymbol{Z} - \boldsymbol{C}$. Fig. 14(b) gives a plot of $\boldsymbol{Z} - \boldsymbol{C}$. Here, we can easily identify the peaks and reduce the noise added by them by replacing it with zero. The peaks can be identified by selecting an appropriate threshold. We set the threshold value to 15 and replace the magnitude of the peaks lying above the threshold with zero. The experimental BER plot after the peak removal is given in Fig. 14(c). From Fig. 14(c), we can see that a BER of the order of $\approx 10^{-7}$ can be achieved.

For SCS-QIM scheme, we will evaluate the effect of the noise added due to round-off process. Let us consider this noise is distributed uniformly between $-\tau$ and τ. For simplicity, we assume the key $K_{qim} = 0$. The watermark can be assumed to be distributed in the interval $(-\beta\Delta, \beta\Delta)$ and the watermarked signal can be written as

$$\begin{aligned} \boldsymbol{C_W} &= \boldsymbol{C} + \beta(\boldsymbol{Q}_\Delta(\boldsymbol{C}) - \boldsymbol{C}) \\ &= \boldsymbol{Q}_\Delta(\boldsymbol{C}) - (1-\beta)(\boldsymbol{Q}_\Delta(\boldsymbol{C}) - \boldsymbol{C}). \end{aligned} \tag{43}$$

So, the received signal can be written as $\boldsymbol{Z} = \boldsymbol{C}_W + \boldsymbol{N}$. Since we assume a flat pdf for host signal within a quantization bin, the probability of error becomes independent to the quantization bin in which the host lies. Thus taking out $Q_\Delta(\cdot)$ does not affect the P_e and defining, $\tilde{z} = n - (1-\beta)(\boldsymbol{Q}_\Delta(\boldsymbol{C}) - \boldsymbol{C})$, then integrating P_e in $[Q_{-1}(c_i) - \tau - (1-\beta)\Delta, Q_{-1}(c_i) + \tau + (1-\beta)\Delta]$

$$P_e = \int f_z(z \,|\, w_j = -1)\, dz = \int f_{\tilde{z}}(\tilde{z}) d\tilde{z} \tag{44}$$

where $f_z(\cdot)$ denotes the pdf of $\boldsymbol{Z}$. The probability of error is then given as [39]

$$P_e = 0, \quad \theta \geq \frac{\beta}{\beta - 1/2} \tag{45}$$

$$= \frac{\beta - (\beta - 1/2)\xi^2}{4\beta(1-\beta)\theta}, \quad \frac{\beta}{3/2 - \beta} \leq \theta < \frac{\beta}{\beta - 1/2} \tag{46}$$

$$= \frac{5/2 - 2\beta}{3/2 - \beta} - \frac{3/2 - \beta}{\beta}\theta, \quad \theta < \frac{\beta}{3/2 - \beta} \tag{47}$$

where $\theta = \beta\Delta/\tau$.

REFERENCES

[1] S. Hwang, K. Yoon, K. Jun, and K. Lee, "Modeling and implementation of digital rights," *J. Syst. Softw.*, vol. 73, no. 3, pp. 533–549, 2004.

[2] A. Sachan, S. Emmanuel, A. Das, and M. S. Kankanhalli, "Privacy preserving multiparty multilevel DRM architecture," in *Proc. 6th IEEE Consumer Communications and Networking Conf., Workshop Digital Rights Management*, 2009, pp. 1–5.

[3] T. Thomas, S. Emmanuel, A. Subramanyam, and M. Kankanhalli, "Joint watermarking scheme for multiparty multilevel DRM architecture," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 4, pp. 758–767, Dec. 2009.

[4] A. Subramanyam, S. Emmanuel, and M. Kankanhalli, "Compressed-encrypted domain JPEG2000 image watermarking," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2010, pp. 1315–1320.

[5] H. Wu and D. Ma, "Efficient and secure encryption schemes for JPEG 2000," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2004, vol. 5, pp. 869–872.

[6] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in *Proc. 11th ACM Workshop Multimedia and Security*, 2009, pp. 9–18.

[7] T. Bianchi, A. Piva, and M. Barni, "Composite signal representation for fast and storage-efficient processing of encrypted signals," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 180–187, Mar. 2010.

[8] S. Lian, Z. Liu, R. Zhen, and H. Wang, "Commutative watermarking and encryption for media data," *Opt. Eng.*, vol. 45, pp. 1–3, 2006.

[9] F. Battisti, M. Cancellaro, G. Boato, M. Carli, and A. Neri, "Joint watermarking and encryption of color images in the Fibonacci-Haar domain," *EURASIP J. Adv. Signal Process.*, vol. 2009.

[10] M. Cancellaro, F. Battisti, M. Carli, G. Boato, F. De Natale, and A. Neri, "A joint digital watermarking and encryption method," in *Proc. SPIE Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, 2008, vol. 6819, pp. 68 191C–68 191C.

[11] J. Prins, Z. Erkin, and R. Lagendijk, "Anonymous fingerprinting with robust QIM watermarking techniques," *EURASIP J. Inf. Security*, vol. 2007.

[12] Z. Li, X. Zhu, Y. Lian, and Q. Sun, "Constructing secure content-dependent watermarking scheme using homomorphic encryption," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2007, pp. 627–630.

[13] Q. Sun, S. Chang, M. Kurato, and M. Suto, "A quantitive semi-fragile JPEG2000 image authentication system," in *Proc. Int. Conf. Image Processing*, 2002, vol. 2, pp. 921–924.

[14] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[15] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[16] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," *Lecture Notes in Computer Science*, pp. 223–238, 1999.

[18] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal Process.: Image Commun.*, vol. 17, no. 1, pp. 3–48, 2002.

[19] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proc. 2nd Annu. Int. Conf. Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2005)*, 2005, pp. 109–117.
[20] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Process.*, vol. 66, no. 3, pp. 283–301, 1998.
[21] J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1003–1019, Apr. 2003.
[22] F. Perez-Gonzalez, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: A high-rate data-hiding method invariant to gain attacks," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pt. 2, pp. 3960–3975, Oct. 2005.
[23] B. Schneier, *Applied Cryptography*. New York: Wiley, 1996.
[24] D. Engel, T. Stutz, and A. Uhl, "A survey on JPEG2000 encryption," *Multimedia Syst.*, vol. 15, no. 4, pp. 243–270, 2009.
[25] G. Paul, S. Rathi, and S. Maitra, "On non-negligible bias of the first output byte of RC4 towards the first three bytes of the secret key," *Designs, Codes, Cryptography*, vol. 49, no. 1, pp. 123–134, 2008.
[26] A. Klein, "Attacks on the RC4 stream cipher," *Designs, Codes, Cryptography*, vol. 48, no. 3, pp. 269–286, 2008.
[27] C. Shannon, "Communication theory of secrecy systems," *MD Comput.*, vol. 15, no. 1, pp. 57–64, 1998.
[28] S. Fluhrer and D. McGrew, "Statistical analysis of the alleged RC4 keystream generator," *Lecture Notes in Computer Science*, pp. 19–30, 2001.
[29] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," *Lecture Notes in Computer Science*, pp. 1–24, 2001.
[30] ETSI/SAGE, Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2, Document 2: SNOW 3G Specification, Version 1.1, 2006.
[31] I. Mantin and A. Shamir, "A practical attack on broadcast RC4," *Lecture Notes in Computer Science*, pp. 152–164, 2002.
[32] W. Trappe *et al.*, "Anti-collusion fingerprinting for multimedia," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 1069–1087, Apr. 2003.
[33] F. Hartung, J. Su, and B. Girod, "Spread spectrum watermarking: Malicious attacks and counterattacks," *Security and Watermarking of Multimedia Contents*, pp. 147–158, 1999.
[34] N. Memon and P. Wong, "A buyer-seller watermarking protocol," *IEEE Trans. Image Process.*, vol. 10, no. 4, pp. 643–649, Apr. 2001.
[35] A. Abrardo, M. Barni, F. Pérez-González, and C. Mosquera, "Improving the performance of RDM watermarking by means of trellis coded quantisation," *IEE Proc. Inf. Security*, vol. 153, no. 3, pp. 107–114, 2006.
[36] A. Weber, The USC-SIPI Image Database Version 5, USC-SIPI Report, 1997, vol. 315, pp. 1–24.
[37] G. Schaefer and M. Stich, "UCID—An uncompressed colour image database," *Multimedia Syst.*, vol. 15, no. 4, pp. 243–270, 2009.
[38] A. Papoulis, S. Pillai, and S. Unnikrishna, *Probability, Random Variables, and Stochastic Processes*. New York: McGraw-Hill, 1965, vol. 196.
[39] F. Pérez-González, F. Balado, and J. Martin, "Performance analysis of existing and new methods for data hiding with known-host information in additive channels," *IEEE Trans. Signal Process.*, vol. 51, no. 4, pp. 960–980, Apr. 2003.

A. V. Subramanyam received the B.Tech. degree in electronics engineering from the Indian School of Mines, Dhanbad, India, in 2007. Currently he is pursuing the Ph.D. degree in the School of Computer Engineering, Nanyang Technological University, Singapore.

His current research interests are in multimedia watermarking and multimedia forensics.

Sabu Emmanuel (M'02) received the B.E. degree in electronics and communication engineering from Regional Engineering College, Durgapur, India, in 1988, the M.E. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1998, and the Ph.D. degree in computer science from the National University of Singapore in 2002.

He is currently an Assistant Professor in the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. His current research interests are in multimedia and software security and surveillance video processing.

Mohan S. Kankanhalli (SM'09) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, and the M.S. and Ph.D. degrees in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY.

He is a Professor in the School of Computing, National University of Singapore. His current research interests are in multimedia systems (content processing, retrieval) and multimedia security (surveillance, authentication, and digital rights management).

Prof. Kankanhalli is on the Editorial Board of several journals, including *ACM Transactions on Multimedia Computing, Communications, and Applications*, *Multimedia Systems Journal*, and *Multimedia Tools and Applications*.