COMPRESSED-ENCRYPTED DOMAIN JPEG2000 IMAGE WATERMARKING

A. V. Subramanyam[†]*, Sabu Emmanuel*[†]*, Mohan S. Kankanhalli*[‡]

[†]School of Computer Engineering, Nanyang Technological University, Singapore
[‡]School of Computing, National University of Singapore, Singapore
Email: {subr0021, asemmanuel}@ntu.edu.sg, mohan@comp.nus.edu.sg

ABSTRACT

In digital rights management (DRM) systems, digital media is often distributed by multiple levels of distributors in a compressed and encrypted format. The distributors in the chain face the problem of embedding their watermark in compressed, encrypted domain for copyright violation detection purpose. In this paper, we propose a robust watermark embedding technique for JPEG2000 compressed and encrypted images. While the proposed technique embeds watermark in the compressed-encrypted domain, the extraction of watermark can be done either in decrypted domain or in encrypted domain.

Keywords— Compressed and Encrypted Domain Watermarking, JPEG2000 Watermarking, Robust.

1. INTRODUCTION

Over the past decade, there has been an explosive growth in the digital media creation/capturing, processing and distribution. The media is often distributed in compressed and encrypted format and watermarking of these media for copyright violation detection, proof of ownership or distributorship, media authentication, needs to be carried out in compressed-encrypted domain. One such example is the distribution through DRM systems [1] [2], where the owner of media, distribute the media in compressed and encrypted format to consumers through multi-level distributor network. In these DRM systems, in order for the distributors to prove their distributorship to consumers or for consumer copyright violation detection, watermarking needs to be performed by the distributors in the compressed-encrypted domain, as they do not have access to the unencrypted media. In this paper we focus on JPEG2000 compressed images. There have been several related image watermarking techniques proposed till date [3]-[7]. Cancellaro et.al. in [3] proposed a joint digital encryption and watermarking scheme, in which most significant bit planes are encrypted while the remaining bit planes are watermarked. In this case the encryption bit planes need to be decided a priori and the rest of the bit planes are in plaintext form for watermarking. Leaving few bit planes in plaintext form may not be good as it might help the attacker to pick the right image to attack. In [4] Lian et.al. proposed a technique of watermarking in a wavelet transformed image. Some of the subbands are encrypted while others are watermarked. This algorithm also suffers from similar weaknesses as that of [3], where the subbands to be encrypted should be decided a priori and since some subbands are in plaintext form, the attacker can pick the right image to attack. Prins et.al. in [5] proposed a robust quantization index modulation (QIM) based watermarking technique, which embeds the watermark in the encrypted domain. The technique makes use of encrypted quantized samples of plain text host signal which renders it unsuitable for watermarking in compressed domain. In [6] Li et.al. proposed a content-dependent watermarking technique, which embeds the watermark in an encrypted format, but the host signal is still in plain text format. The algorithm may not be directly applied when the content is in encrypted format, in that case the distortion introduced in the host signal may be large. In [7] Sun et.al. proposed a semi fragile authentication system for JPEG2000 images. However, this scheme is not a fully compressed and encrypted domain watermarking compatible as it derives the content based features for watermarking from the plain text. To our knowledge, the proposed technique is the first work which does fully compressed encrypted domain watermarking. The paper is organized as follows. In section 2, we give the challenges of compressed-encrypted domain watermarking. Section 3 describes the proposed scheme. In section 4, we discuss the key distribution and security analysis of encryption and watermarking algorithm. The experimental results are discussed in section 5. Section 6 concludes the paper.

2. CHALLENGES OF COMPRESSED-ENCRYPTED DOMAIN WATERMARKING

1) **Compressed domain watermarking**: The distributed content is generally in compressed and encrypted format. Once encrypted by the owner, the content cannot be decompressed at any of the intermediate distributor levels without decrypting it. Therefore, it should be possible to embed the watermark in the compressed byte stream. However, 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 perceptual quality of image is minimal.

2) **Encrypted domain watermarking and watermark re-**

We thank the Agency for Science, Technology and Research (A*STAR), Singapore for supporting this work under the project "Digital Rights Violation Detection for Digital Asset Management" (Project No: 0721010022).

978-1-4244-7493-6/10/$26.00 ©2010 IEEE

trieval: Once the distributor receives the compressed-encrypted content, it should be possible to embed the watermark without requiring the decryption of content. This is required to preserve the confidentiality of content and the cryptosystem should also be secure enough to resist different attacks to decrypt the cipher. Also it is desired that the watermark should persist under some obvious manipulations such as content decryption i.e. the watermark detection should be possible, even after the decryption. Therefore, the cryptosystem must be privacy homomorphic, explained in section 3.1, in order to detect the watermark correctly. Symmetric stream cipher with homomorphic property is preferred over asymmetric encryption with homomorphic property mainly due to the following two reasons. Firstly, if the encryption is performed using homomorphic asymmetric schemes, like RSA [8], Goldwasser-Micali [9], Elgamel [10] and Paillier [11], 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'). The size of ciphertext in case of Paillier and Elgamel cryptosystem is twice that of plaintext [11] [10]. 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. Symmetric ciphers with homomorphism, on the other hand, 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 of deciding which cipher scheme to be applied.

3. PROPOSED SCHEME

The proposed algorithm works on JPEG2000 compressed code stream. In JPEG2000 encoder, the 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. The compressed byte stream is arranged into different wavelet packets based on resolution, precincts, components and layers. 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. We use Hartung et.al.'s [12] spread spectrum technique for the purpose and study the bit error rate of detection and the quality versus payload capacity trade-off. The watermark detection is done after the decryption but in the compressed domain as shown in Figure 1. Figure 1 shows the watermark embedding and detection pipelines. We explain encryption algorithm next.

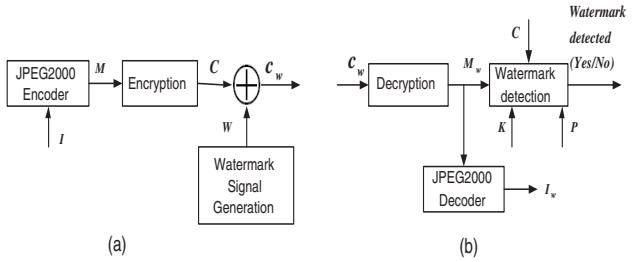

Fig. 1. (a). Watermark embedding (b). Watermark extraction

3.1. Encryption Algorithm

JPEG2000 gives out packetized byte stream M as its output, where $M = \{m_i\}$, $m_i \in$ [0,255] $\forall\ i$ =0, 1,...,$L-1$. L is the length of the message in bytes. In order to encrypt the message M, we choose K, a randomly generated key-stream using RC4, explained in section 4.1, within the set $K = \{k_i\}$ where $k_i \in$ [0,254] $\forall\ i$ =0, 1,...,$L-1$. Then the encryption is done byte by byte as given in equation (1) to get the ciphered signal C.

$$C = E(M, K) = c_i = (m_i + k_i)\ mod\ 255\ \forall i = 0, 1, ..., L-1 \quad (1)$$

where $E(.)$ is the encryption function and the addition operation is arithmetic addition. Let $C_1 = E(M_1, K_1)$ and $C_2 = E(M_2, K_2)$. For $K = K_1 + K_2$, additive homomorphism property gives,

$$D(C_1 + C_2, K) = M_1 + M_2 \quad (2)$$

Thus this stream cipher has additive privacy homomorphism property [13]. 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. Security of this stream cipher is elaborated in section 4.2. Distributors in the distribution chain are given this compressed encrypted byte stream C to distribute. They do not have access to the original image. Often distributors need to watermark C to prove their distributorship to the recipient or copyright violation detection purposes. Next we explain the watermarking algorithm.

3.2. Embedding Algorithm

Since the encryption algorithm is with additive privacy homomorphism property, any robust additive watermarking technique can be used. We make use of Hartung et.al.'s [12] spread spectrum technique for the purpose. 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, modifying which 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 the

watermark can be inserted in less significant bit planes of middle resolutions without affecting the image quality much. Let the watermark information bits be $\mathbf{b} = \{b_i\}\ \forall\ i$=0, 1,..., $N-1$, where $b_i \in$ {-1,1}.This is then spread by a factor chip rate r, which gives

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

The sequence a_j is then multiplied by a strength factor $\alpha > 0$ and PN sequence $P=\{p_j\}$ with zero mean and variance σ_p^2, where $p_j \in$ {-1,1}. The watermark signal $W=\{w_j\}$, where,

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

The watermark signal generated in equation (4) is added to the encrypted signal C, to give the watermarked signal C_W,

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

Thus, watermark embedding is carried out in compressed-encrypted domain. However, it is shown in section 3.3 that the watermarked quality can be controlled in a predictable manner. Distributor distributes the compressed-encrypted watermarked image.

3.3. Watermark Detection

The received compressed-encrypted watermarked image is first decrypted using the equation (6), which defines the corresponding byte by byte decryption for the encryption defined in equation (1). The keystream K can be generated as given in section 4.1. Let the watermarked image be denoted as $M_W = \{m_{w_i}\}$ where $m_{w_i} \in$ [0,255] $\forall\ i$ =0, 1,...,$L-1$. Then the received signal C_W is decrypted to give M_W as,

$$\begin{aligned} M_W = D(C_W, K) &= (c_{w_i} - k_i)\ mod\ 255\ \ \forall i = 0, 1, ..., L-1 \\ &= (c_i + w_i - k_i)\ mod\ 255 \\ &= m_i + w_i \\ &= m_{w_i} \end{aligned} \tag{6}$$

where $D(.)$ is the decryption function. It can be seen from equation (6) 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. The embedded watermark information W can be estimated from M_W or C_W using correlation detector even without the knowledge of the corresponding originals M or C. Incase of M_W it is multiplied by PN sequence P used for embedding, followed by summation over chip-rate window r, yielding the correlation sum S_i.

$$\begin{aligned} S_i &= \sum_r (m_{w_j} p_j) \\ &= \sum_r (m_j + w_j) p_j \\ &= \sum_r m_j p_j + \sum_r w_j p_j \end{aligned} \tag{7}$$

The first term in equation (7) is zero if M and P are uncorrelated. However, this is not always the case for real data. To obtain better detection results, we can pre-filter M_W and remove most of the image content. Assuming that the first term in equation (7) is zero

$$S_i = \sum_r (w_j p_j) = \sum_r \alpha a_j p_j p_j = b_i \sigma_p^2 \alpha r \tag{8}$$

Thus, the sign of S_i gives the watermark information bit

$$sign(S_i) = sign(b_i \sigma_p^2 \alpha r) = sign(b_i) = b_i \tag{9}$$

It is also possible to detect the watermark in the compressed-encrypted watermarked image C_W. Since $C_W = C + W$, the correlation detector can be applied to C_W instead of M_W. Thus, assuming zero correlation between C and P.

$$S_i = \sum_r (c_{w_j} p_j) = \sum_r (c_j + w_j) p_j = b_i \sigma_p^2 \alpha r \tag{10}$$

Here again, the sign of S_i gives the watermark information bit as derived in equation (9). Incase of copyright violation detection purpose, since the distributors have C, they can apply the non-blind detection technique, i.e., subtract away C from C_W to remove the correlation effect completely. Thus get a better watermark detection rate. However, the distributors can also use pre-filtered (semi-blind) detection technique. Incase of ownership proving applications the pre-filtered (semi-blind) detection technique may be required. The security of this spread spectrum technique is discussed in section 4.3.

4. DISCUSSION

4.1. Key Stream Generation

The keystream is generated at the encryption and decryption site using RC4 cipher [14]. For encryption, a secret seed S is applied to RC4 cipher which in turn generates the keystream K. In order to generate the same key K at the decryption site, the seed S must be delivered to the decryption site through a secret channel. Once the seed S is received, it can be applied to RC4 cipher to generate the key stream K.

4.2. Security Analysis

The security of proposed encryption algorithm relies significantly on the security of the underlying technique used in the specific cryptosystem. The RC4 cryptosystem that we have

used, is a very well known technique and is believed to be secure with the discarding of atleast first few hundred bytes [15]. Henceforth, we will continue the discussion assuming this result being applied. Here, we discuss the security of the applied encryption scheme, which is based on Shannon's theory of security [16]. We assume that $M \sim U[0, 255]$ and let P_M and P_K denote the probabilities of occurrence of random variables $\boldsymbol{K}$ and $\boldsymbol{M}$ respectively, so the amount of information contained in M is given by,

$$H_M = -\sum P_M \log P_M \tag{11}$$

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

$$H_{K'} = -\sum P_{K'} \log P_{K'} \tag{12}$$

For perfect secret systems the amount of information H_M can be hidden completely if $H_{K'} >= H_M$ [16], which is correct if the key is truly random. However, the RC4 keystream K is a pseudo-random sequence having a different distribution than that of a random keystream 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 [17] Fluhrer et. al. derived that, it requires $\approx 2^{30.6}$ bytes to discriminate RC4 output cipher from a truly random sequence. 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 S is used more than once. In this case, the seed S is sent as a concatenation of S with a key (referred to as Initialization Vector IV, need not be secret). This attack tries to find the secret seed S by observing the output streams for different IV values and a fixed S, and can recover the keystream K without much time complexity [18]. However, it can be overcome by using different secret keys, in which case the attack scenario is same as [17]. In [19] Mantin et. al. proposed an attack by observing first two output words of the cipher and deduced 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 and it becomes insignificant for output word size of more than 5 bits.

4.3. Security of Watermarking Algorithm

The security and robustness of the watermarking algorithm depends on the underlying spread spectrum watermarking technique. The attacks to retrieve or destroy the watermark can be performed either in encrypted or decrypted compressed domain. However, attacks in encrypted domain may result into a random decrypted stream which degrades the image quality. Hence, decrypted-compressed content provides a better domain to attack. The watermark detection performance against attacks like additive Gaussian noise, filtering, and amplitude scaling is given in section 5. The algorithm is robust against additive Gaussian noise as can be seen from Figure 5. The robustness against filtering, such as 1x5 median filter and scaling attack can be improved by increasing the chip rate and estimating the scale factor [20] respectively. The scale factor can be measured accurately using the data before and after attack. Collusion attacks can be made ineffective by using collusion resistant codes can be used to identify all or atleast groups of users involved in collusion [21]. Some attacks may try to render ownership proving 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 [22]. Thus the party which embeds its watermark first in the content can provide the hash of the original content and identifier to detect his own watermark. However, the party which embeds subsequently has a watermarked content and cannot provide hash of original content and identifier and cannot detect his own watermark in the original content.

5. EXPERIMENTAL RESULTS

Experiments are carried out for 30 grayscale images with dimensions 512x512, numbers of compression resolutions are 6, and compression rate is 16. We use scaling factor $\alpha = 3$ and chip-rate r = 32 for the experiments for good robustness against attacks. Further, we show the relation between the number of watermarked bit planes, the number of resolutions used for watermarking, payload capacity and bit error rate against different attacks. Figure 2 gives a measure of the average payload capacity vs number of bit planes(l) watermarked under different resolutions. The average size of the compressed image is 15,901 bytes. Average payload capacity is given here as the ratio of the average embedded number of bits to the average compressed stream size (in bytes) in percentage, where average is computed as a simple mean. In the Figure 2, the payload capacity for the number of watermarked bitplanes $= l$ means the bitplanes $l, l-1, .., 1$ are watermarked. For $l = 8$, means all the bitplanes are watermarked, $l = 7$ means all the bitplanes except the most significant bitplane are watermarked. Similarly, $l = 1$ means only the least significant bitplane is watermarked. 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. Although resolution 4(HL2, LH2, HH2) gives more capacity than resolution 5(HL1, LH1, HH1), it is due to the rate-distortion optimization 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 resolution 5 is less than that of resolution 4. Figure 3 shows PSNR against l for the payload given in Fig-

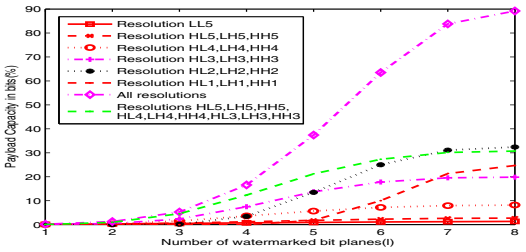

Fig. 2. Average Payload Capacity vs. Number of Watermarked Bit Planes(l)

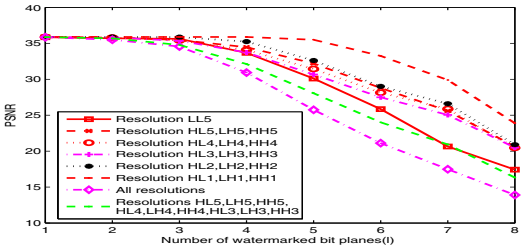

Fig. 3. Average PSNR Vs Number of Watermarked Bit Planes(l)

ure 2, where PSNR is calculated as,

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j) - I_w(i,j))^2 \tag{13}$$

$$PSNR = 10\log_{10}(255^2/MSE) \tag{14}$$

Mean square error (MSE), is the sum of squares of difference between the original image I and watermarked decompressed image I_w. The degradation of image quality due to watermarking lower resolutions especially LL5 is more than caused by watermarking higher resolutions, this effect is more prominent for $l > 4$. This is because the higher resolutions does not carry much of the relevant data, 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 $l = 4$ of middle resolutions provide a good region for watermarking. The average payload capacity for $l = 4$, considering all the resolutions, is 2473 bits with an average PSNR of 30.93 dB, whereas when resolutions 1, 2 and 3 are considered the average capacity is 1962 bits with an average PSNR of 32.12dB, for a compressed image of average size 15,901 bytes. Figure 4 shows the original image, unwatermarked-decompressed image, encrypted image and watermarked-decompressed image. The performance of the watermarking scheme against noise is given in Figure 5 for non-blind detection. From the Figure 5, it is clear that, the

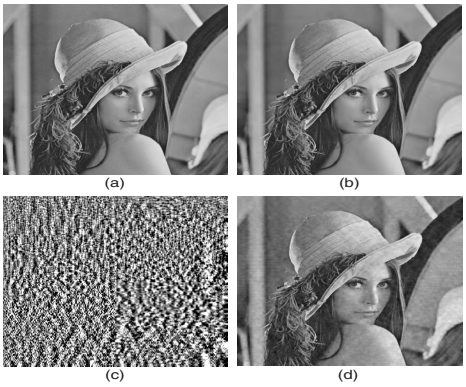

Fig. 4. (a) Original image (b) Unwatermarked-Decompressed image (40.22dB) (c) Encrypted image (12.48dB) (d) Watermarked-Decompressed image ($l = 4$, Payload=2464 bits, PSNR=34.48dB)

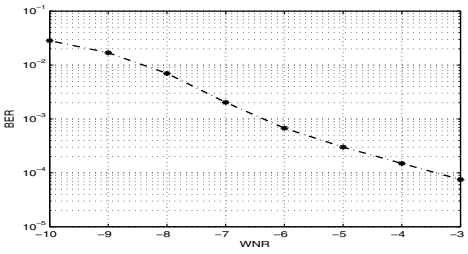

Fig. 5. BER vs. WNR

scheme gives good results even at lower watermark to noise ratio (WNR), where WNR is,

$$WNR = 10\log(P_W/P_N) \tag{15}$$

where, P_W and P_N denotes the watermark power and noise power respectively. 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 only on the amount of noise added. For WNR>-3db, their is no error in watermark retrieval. BER for 1x5 median filtering is 3.38×10^{-3} for a chip rate of 200. The watermark can be detected without any bit errors for scaling attack as the scale factor is estimated accurately.

6. CONCLUSION

In this paper we proposed a technique to embed a robust watermark in the JPEG2000 compressed encrypted images. The algorithm is simple to implement as it is directly performed on 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 properties of the cryptosystem are exploited, which allows us to detect the watermark after decryption and control the image quality as well.

We analyze the relation between payload capacity and quality of the image 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. We also study BER and the results show that the proposed technique performs well against attacks like additive Gaussian noise, filtering, and amplitude scaling.

7. REFERENCES

[1] S.O. Hwang, K.S. Yoon, K.P. Jun, and K.H. Lee, "Modeling and implementation of digital rights," *The Journal of Systems & Software*, 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 *6th IEEE Consumer Communications and Networking Conference, Workshop on Digital Rights Management, 2009.*, 2009, pp. 1–5.

[3] M. Cancellaro, F. Battisti, M. Carli, G. Boato, FGB De Natale, and A. Neri, "A joint digital watermarking and encryption method," in *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X. Edited by Delp, Edward J., III; Wong, Ping Wah; Dittmann, Jana; Memon, Nasir D. Proceedings of the SPIE*, 2008, vol. 6819, pp. 68191C–68191C.

[4] S. Lian, Z. Liu, R. Zhen, and H. Wang, "Commutative watermarking and encryption for media data," *Optical Engineering*, vol. 45, pp. 080510.1–080510.3, 2006.

[5] JP Prins, Z. Erkin, and RL Lagendijk, "Anonymous fingerprinting with robust QIM watermarking techniques," *EURASIP Journal on Information Security*, vol. 2007, no. 8, 2007.

[6] Z. Li, X. Zhu, Y. Lian, and Q. Sun, "Constructing Secure Content-Dependent Watermarking Scheme using Homomorphic Encryption," in *IEEE International Conference on Multimedia and Expo*, pp. 627–630, 2007.

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

[8] RL Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 126, 1978.

[9] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[10] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT'99*, 1999.

[12] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, 1998.

[13] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, 2005, pp. 109–117.

[14] Bruce Schneier, *Applied Cryptography*, John Wiley and Sons, New York, 1996.

[15] 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 and Cryptography*, vol. 49, no. 1, pp. 123–134, 2008.

[16] C.E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[17] S. Fluhrer and D. McGrew, "Statistical analysis of the alleged RC4 keystream generator," in *Fast Software Encryption*, 2000.

[18] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the key scheduling algorithm of RC4," in *Selected Areas in Cryptography*, 2001.

[19] I. Mantin and A. Shamir, "A practical attack on broadcast RC4," in *Fast Software Encryption*, 2001.

[20] J.J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1003–1019, 2003.

[21] W. Trappe, M. Wu, Z.J. Wang, K.J.R. Liu, et al., "Anti-collusion fingerprinting for multimedia," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 1069–1087, 2003.

[22] F. Hartung, J.K. Su, and B. Girod, "Spread spectrum watermarking: Malicious attacks and counterattacks," *in Proc. SPIE Security and Watermarking of Multimedia Contents 99*, pp. 147–158, 1999.