CONTEXT BASED WATERMARKING OF SECURE JPEG-LS IMAGES

A. V. Subramanyam and Sabu Emmanuel
School of Computer Engineering, Nanyang Technological University, Singapore, Republic of Singapore

Keywords: Compressed domain watermarking, Context based watermarking, JPEG-LS watermarking.

Abstract: JPEG-LS is generally used to compress bio-medical or high dynamic range images. These compressed images sometime needs to be encrypted for confidentiality. In addition, the secured JPEG-LS compressed images may need to be watermarked to detect copyright violation, track different users handling the image, prove ownership or for authentication purpose. In the proposed technique, watermark is embedded in the *context* of the compressed image while the Golomb coded bit stream is encrypted. The extraction of watermark can be done during JPEG-LS decoding. The advantage of this watermarking scheme is that the media need not be decompressed or decrypted for embedding watermark thus saving computational complexity while preserving the confidentiality of the media.

1 INTRODUCTION

The digital media is often embedded with watermarks for various purposes such as fingerprinting, copyright violation detection, proof of ownership, authentication and tamper proofing. And is often distributed in a secured manner, for e.g in Digital Rights Management (DRM) Systems or Clinical Information Systems (CIS). High dynamic range images in DRM systems, and Biomedical images in CIS systems are generally compressed using JPEG-LS. JPEG-LS is commonly used to compress these images as the compression efficiency is better and also facilitates near-lossless compression than JPEG and JPEG2000 (Fang, 2009).

In DRM systems, the media providers often compress the media and distribute it after encrypting the compressed media via multilevel distributors (Thomas et al., 2009). In the distribution process the media is transmitted from owners to the consumers through different level of distributors. In this scenario, the distributors are entitled only to distribute the compressed encrypted media to the end user and as such cannot access the plain content (un-encrypted content). Distributors request the license server in the DRM system to distribute the associated licence containing the decryption keys to open the encrypted content to the consumers. However, each distributor sometime needs to watermark the content for traitor tracing or proving the distributorship. Thus they need to watermark in the compressed encrypted domain itself.

In biomedical field, CIS often manages the patient related media in a compressed and confidential way in different *Healthcare Establishments* (Blobel, 2004). In such a communication, different patient records can be shared between different professionals either for subject's treatment purpose or for a case study. Thus it becomes necessary to preserve the confidentiality of the distributed media, to prove the ownership and also track the user who is dealing with the record. Therefore the watermark needs to be inserted in the secured compressed media itself.

In this paper we focus on robust watermarking for copyright violation protection and fingerprinting. Although a robust irreversible watermark can be used for this purpose but it degrades the quality of the image which is not desirable in the applications specified above. Therefore it should be watermarked in such a domain which does not create any undesirable distortion in the image. In literature some of the irreversible algorithms have been proposed for encrypted domain watermarking in general, and also for watermarking of JPEG-LS. In (Caldelli et al., 2006), Caldelli et. al. proposed a watermarking scheme for authentication using the prediction error of the pixels during JPEG-LS encoding . Some techniques have been proposed where watermarking is done on cer-

Subramanyam A. and Emmanuel S..
CONTEXT BASED WATERMARKING OF SECURE JPEG-LS IMAGES.
DOI: 10.5220/0003446201610166
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications* (SIGMAP-2011), pages 161-166
ISBN: 978-989-8425-72-0

tain subbands/bitplanes while encrypting certain other subbands/bitplanes (Cancellaro et al., 2008), (Lian et al., 2006). Some of the encrypted domain algorithms are proposed in (Bianchi et al., 2010), (Piva et al., 2010), (Zhao et al., 2010), (Katzenbeisser et al., 2008). However, these algorithms cause irreversible distortion which might not be desirable especially in case of biomedical images. Thus, here we propose a technique for watermarking of encrypted JPEG-LS compressed images, where the compressed bitstream is encrypted while watermark is embedded using *contexts* (explained in section 2). This paper is organized as follows. Section 2 gives the preliminaries. In section 3, we present the embedding and detection process. We discuss the experimental results in section 4. Section 5 concludes the paper.

2 PRELIMINARIES

In this section we briefly discuss the JPEG-LS compression algorithm (Weinberger et al., 2000), and the challenges in watermarking a compressed-encrypted JPEG-LS image. In figure 1-a, an *initialcontext* is computed in the *Gradient* block using the pixels a, b, c and d. This *initialcontext* is a positive quantity and is mapped into equal negative and positive values using a *classmap*. For e.g., let the *initialcontext* be denoted as $\boldsymbol{Q} \in [0,728]$ (Weinberger et al., 2000). Let us denote $\boldsymbol{Q_{neg}} \in [-1,-364]$ and $\boldsymbol{Q_{pos}} \in [1,364]$ and let *classmap* $f(.)$ denote the function which maps $\boldsymbol{Q}$ to $\boldsymbol{Q_{neg}}$ or $\boldsymbol{Q_{pos}}$ or 0, i.e, $f : \boldsymbol{Q} \rightarrow \{\boldsymbol{Q_{pos}}, \boldsymbol{Q_{neg}}, 0\}$. In the *regular* mode, the current sample x is predicted in *Predictor* block using a finite past pixel set comprising of pixels a, b and c. Generally the fixed prediction has some bias in it which is canceled out using the *context* in which pixel occurs. Finally, the prediction residual between the corrected (bias canceled) predicted value and the original pixel is Golomb coded. Further each pixel (of a M x N image) occurs under $\boldsymbol{Q}_{px[j]} \in \{\boldsymbol{Q_{pos}}, \boldsymbol{Q_{neg}}, 0\}$ while encoding, where $px[j]$ denotes the pixel at position j, $\forall$ j = 1, 2, ..., M x N and $\boldsymbol{Q}_{px[j]}$ denotes its *context*. Also, the Golomb coded bit stream can be encrypted using a secure cipher scheme such as RC4 (Schneier, 1996) for confidentiality, which can then be distributed.

Now, modifying such a randomized bitstream to insert watermark faces certain *limitations*. The Golomb coded bitstream is highly sensitive to bit errors. Even if one bit gets corrupted, the rest of the decoding can be compromised. Therefore, we choose the *context* of the encoded image for embedding watermark without requiring any decryption or decompression. Next we discuss the proposed algorithm.

3 PROPOSED ALGORITHM

The proposed algorithm involves encryption of Golomb coded bit stream while embedding watermark through context. Encryption is performed on the output bit stream of JPEG-LS compression (figure 1-a). The encryption algorithm that we propose to use is RC4 cipher (Schneier, 1996). The encryption does not result in any increase in compressed file size as the encryption is done using a stream cipher. The encrypted bit stream along with *classmap f* and *context* frequencies is sent to the watermark embedder (figure 1-b) where the watermark embedding takes place.

The first step in watermarking process involves finding the frequency or number of occurrences of each *context* $Q \in Qpos$. Then according to the dynamic range of highest frequency *contexts*, the watermark signal is divided into smaller length segments. Further Euclidean distance between the selected *contexts* against the watermark segments is computed. The *contexts* which give minimum distance will represent watermark information. Finally *contexts* involved in computing Euclidean distance are exchanged with the other *contexts* which does not belong to highest occurring *contexts*. The main purpose behind selecting highest occurring contexts is that the higher the dynamic range, the more number of watermark signal bits can be embedded. We describe the watermark embedding procedure next.

3.1 Embedding Process

Let $\boldsymbol{Q_{p_j}}$ $\forall$ $j = 0,1,....,L-1$ be the L highest occurring *contexts* which are selected from $\boldsymbol{Q_{pos}}$ for Euclidean distance criteria. Here, L is chosen such that $L \leq max(\boldsymbol{Q_{pos}})/2$, for e.g. in our case $max(\boldsymbol{Q_{pos}}) = 364$. Let $\boldsymbol{freq}_j$ represent the frequency of $\boldsymbol{Q_{p_j}}$ $\forall$ $j = 0,1,....,L-1$. Now, let us arrange $\boldsymbol{freq}_j$ $\forall$ $j = 0,1,....,L-1$ in ascending order as shown in figure 2 and different $\boldsymbol{freq}_j$'s are then grouped in range R_i, where range R_i $\forall$ $i = 0,1,....,K-1$ represent the i^{th} range, such that $K \leq L$. Although it would be easy to range the frequencies in uniform intervals, this may not be a good idea as it will increase detection error rate as the dynamic range of frequencies in range will vary too much. Therefore, we choose unequal range lengths, with range limits $[\boldsymbol{freq}_{R_i}, \boldsymbol{freq}_{R_{i+1}} - 1] \in R_i$ $\forall$ $i = 0,1,....,K-1$. Let b_i $\forall$ $i = 0,1,....,K-1$ be the minimum number of bits used to represent the elements present in i^{th} range R_i $\forall$ $i = 0,1,....,K-1$. Let their be E_i number of frequency elements (and hence number of *contexts*) in i^{th} range R_i $\forall$ $i = 0,1,....,K-1$. The values of the parameters K and b_i used for simulation purpose are

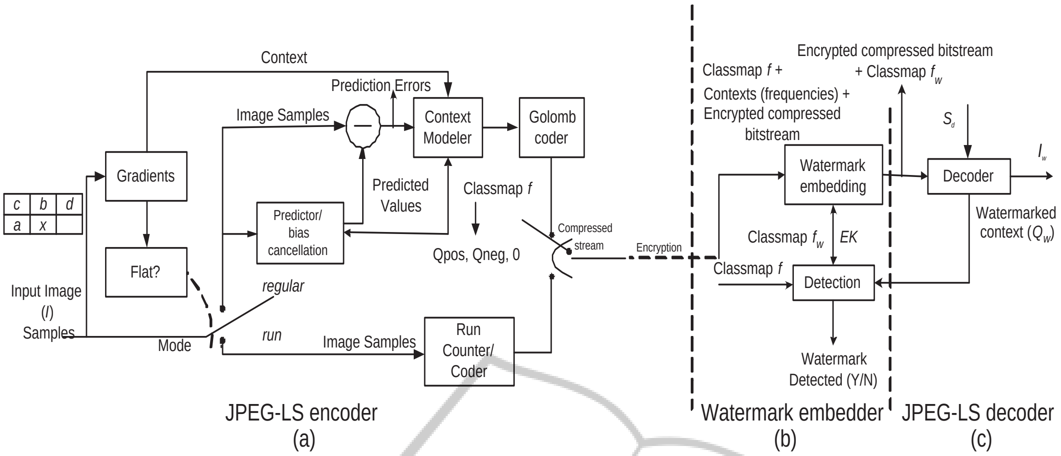

Figure 1: (a)JPEG-LS encoding (b) Watermark embedding/detection (c) Decoding (watermarked context generation).

given in section 4.

Further watermark signal $\boldsymbol{W}$ of length L_w is also segmented. Corresponding to the number of frequency elements in i^{th} range R_i, the length E_ib_i of watermark bits is divided into E_i number of segments, denoted by $w_{ij}\ \forall\ j=0,1,...,E_i-1$, with each segment of length b_i bits. This is done corresponding to each range of figure 2. Also the watermark length L_w is chosen such that,

$$L_w = \sum_{i=1}^{K-1} b_iE_i \tag{1}$$

Now the watermark can be represented as $\boldsymbol{W} = w_{00}||...||w_{ij}\ \ \forall\ i=0,1,....,K-1; j=0,1,...,E_i-1$. Then the Euclidean distance between each watermark segment $w_{ij}\ \forall\ i=0,1,....,K-1; j=0,1,...,E_i-1$ is computed against frequencies of the selected contexts.

$$argmin_{ij,k}(w_{ij},\boldsymbol{freq}_k)\ \forall\ k=0,1,....,L-1 \tag{2}$$

$$\boldsymbol{min}_{Qorder_k} = \boldsymbol{Q}_{\boldsymbol{p}_k} \tag{3}$$

where, $\boldsymbol{min}_{Qorder}$ represents the context whose frequency is closest to the watermark segment. Now, the contexts represented in $\boldsymbol{Q_p}$ are exchanged with the contexts which are not involved in Euclidean distance criteria for watermarking. Now without loss of generality, let $\boldsymbol{Q'_p}$ and $\boldsymbol{Q''_p}$ denote these contexts such that $\boldsymbol{Q_p} \bigcup \boldsymbol{Q'_p} \bigcup \boldsymbol{Q''_p} = \boldsymbol{Q_{pos}}$. Also let $\boldsymbol{Q_n}$, $\boldsymbol{Q'_n}$ and $\boldsymbol{Q''_n}$ denote the corresponding negative contexts of $\boldsymbol{Q_p}$, $\boldsymbol{Q'_p}$ and $\boldsymbol{Q''_p}$ i.e., $\boldsymbol{Q_n} = -\boldsymbol{Q_p}$, $\boldsymbol{Q'_n} = -\boldsymbol{Q'_p}$ and $\boldsymbol{Q''_n} = -\boldsymbol{Q''_p}$ respectively, such that $\boldsymbol{Q_n} \bigcup \boldsymbol{Q'_n} \bigcup \boldsymbol{Q''_n} = \boldsymbol{Q_{neg}}$. Also $\boldsymbol{Q_p}$ and $\boldsymbol{Q'_p}$, and, $\boldsymbol{Q_n}$ and $\boldsymbol{Q'_n}$ are chosen equilength.

In JPEG-LS encoding, *initialcontext* $\boldsymbol{Q}$ is mapped through f as, $f:\boldsymbol{Q} \rightarrow \{Q_p,\boldsymbol{Q'_p},\boldsymbol{Q''_p},\boldsymbol{Q_n},\boldsymbol{Q'_n},\boldsymbol{Q''_n},0\}$. Then, to embed watermark, this mapping is changed by changing f to f_w such that, $f_w:\boldsymbol{Q} \rightarrow \{\boldsymbol{Q'_p},\boldsymbol{Q_p},\boldsymbol{Q''_p},\boldsymbol{Q'_n},\boldsymbol{Q_n},\boldsymbol{Q''_n},0\}$. The mapping can be done as, for e.g., $\boldsymbol{Q'_{p_j}} \leftrightarrow \boldsymbol{min}_{Qorder_j}\ \forall\ j=0,1,....,L-1$, where '$\leftrightarrow$' denotes one-to-one mapping i.e., each element of $\boldsymbol{Q'_{p_j}}$ is mapped to an element of $\boldsymbol{min}_{Qorder_j}$. However, this mapping is secret, can be randomized using a secret embedding key EK, and does not affect the watermarked image quality (explained in section 4.2). Thus $\boldsymbol{Q_p}$ occurring during encoding is replaced with $\boldsymbol{Q'_p}$ for watermarking. $\boldsymbol{Q_n}$ and $\boldsymbol{Q'_n}$ should also be exchanged correspondingly in order to maintain the relation between $\boldsymbol{Q_{pos}}$ and $\boldsymbol{Q_{neg}}$ ((Weinberger et al., 2000)). Since only the classmap f and context frequencies are used for embedding, the encrypted bit stream remains intact. The compressed-encrypted bit stream along with f_w is then sent to the consumer (figure 1-c). Next we discuss the detection algorithm.

3.2 Detection Process

The detection is done while decoding the Golomb coded bit stream. The context map f_w which is sent to the end user generates watermarked context while decoding. The extraction involves f_w, EK, $\boldsymbol{min}_{Qorder}$ and the decryption key. Then the frequencies of the exchanged *contexts* in place of *contexts* occurring in $\boldsymbol{min}_{Qorder}$ are computed, which are then correlated against different watermarks. The original watermark gives the highest correlation, in case if it is present.

Now, $\boldsymbol{min}_{Qorder}$ gives us the order in which the contexts represent the watermark segments. Using EK the *context* mapping is retrieved and, f_w gives us the corresponding context which should be looked for in place of the contexts present in $\boldsymbol{min}_{Qorder}$. Using f_w (which gives the mapping of the *contexts*), the watermarked contexts can be retrieved as, $\boldsymbol{Q}_{\boldsymbol{w}_i}$: $\boldsymbol{min}_{Qorder_i} \rightarrow \boldsymbol{Q'_{p_i}}\ \forall\ i=0,1,....,L-1$ i.e, searching

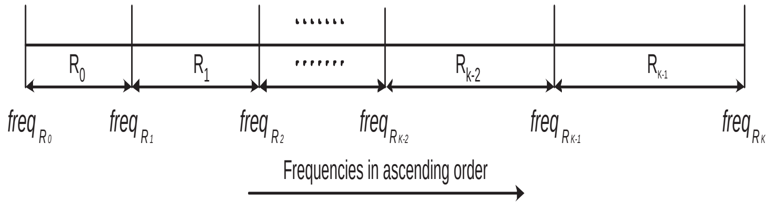

Figure 2: Frequencies arranged in ascending order and grouped.

for the *contexts* in $\boldsymbol{Q'_p}$ which are mapped to *contexts* in $\boldsymbol{min}_{Qorder}$. The frequencies of $\boldsymbol{Q_w}$ in the watermarked *contexts* will then represent the frequencies of $\boldsymbol{min}_{Qorder}$ in the original *contexts*. Let the estimated watermark be $\hat{W} = \hat{w}_0||...||\hat{w}_k \; \forall \; k = 0,1,....,L-1$, where $\hat{w}_k$ is given by,

$$\hat{w}_k = \boldsymbol{freq}_{Q_{w_k}} \; \forall \; k = 0,1,....,L-1 \quad (4)$$

The detection involves correlating the estimated watermark $\hat{W}$ against different watermarks $\check{W}_i \; \forall \; i = 1,2,....,n_w$, where n_w denotes the number of watermarks. Correlation is given by,

$$corr(\hat{W},\check{W}_i) = \frac{E[(\hat{W}-\mu_{\hat{W}})(\check{W}_i-\mu_{\check{W}_i})]}{\sigma_{\hat{W}}\sigma_{\check{W}_i}} \; \forall \; i = 1,....,n_w \quad (5)$$

where corr(.,.) denote the correlation measure, E[.] denote the expectation operator, μ denote the mean, σ denote the variance. The correlation value is then subjected to a threshold T (explained in section 4) to detect the presence of watermark.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

Experiments are carried out on gray scale biomedical images. The parameters K = 5, b_1 = 4, b_2 = 5, b_3 = 6, b_4 = 8 and b_5 = 10 are used. These parameters are selected based on the frequency range and number of contexts occurring in each group. The b_i is chosen to maximize the capacity and minimize the detection error rate. In Table I, the embedding capacity, PSNR, SSIM, and side information for images of different resolutions is given.

4.1 Embedding Capacity

It can be seen from the Table I that the capacity increases with increasing image size. This happens because the larger the image, the higher the frequency of occurrence of *contexts*. And the higher the frequency, more number of bits are required to represent as compared to lesser frequency. However, the image Brain gives lesser capacity, this is because the occurrence of all the *contexts* is uniform as compared to our assumption of highest occurring frequencies used for watermarking. This leads to lesser number of bits required for representing the frequencies of the *contexts* involved in Euclidean distance criteria and hence less capacity.

4.2 Watermarked Image Quality

The image quality does not get affected by the watermarking process which is explained as follows. Since the *context* is used for *bias* cancelation, section 2, we analyze the affect of *context* exchange on bias estimation and establish the fact that, changing the *context* according to the proposed scheme does not affect bias estimation and hence decoded image quality. The bias is estimated based on prediction errors accumulated in the *context*. Let at position px_j a pixel x occurs in *context A* with accumulated error A_{pe}, and B_{pe} be the accumulated error in *context* B. Let the *contexts* A and B occur A_f and B_f times respectively until this point. Now, the bias is given as

$$Bias_A = \lceil A_{pe}/A_f \rceil \quad (6)$$

$$Bias_B = \lceil B_{pe}/B_f \rceil \quad (7)$$

The bias for the predicted value of pixel x is canceled as

$$\hat{x} = \hat{x} + Bias_A \quad (8)$$

where $\hat{x}$ represent predicted value of pixel x.

Let us now calculate the bias in case when the *contexts* are exchanged i.e., A is exchanged with B. In this case, B occurs A_f times while, A occurs B_f times. The prediction error accumulated remains same as that in case of no exchange. Now, the bias is given as,

$$Bias_B = \lceil A_{pe}/A_f \rceil \quad (9)$$

$$Bias_A = \lceil B_{pe}/B_f \rceil \quad (10)$$

The bias cancelation is now given as,

$$\hat{x} = \hat{x} + Bias_B \quad (11)$$

Table 1: Image, Resolution, Embedding capacity, PSNR, SSIM, Side information.

(percentage of		(in bits)			compressed filesize)
Chest1	373 x 387	1147	53.42	.9981	.148
Abdominal	636 x 614	1735	53.33	.9913	.038
Brain	800 x 600	1426	53.59	.9862	.078
Chest2	2048 x 2494	2448	52.84	.9992	.003

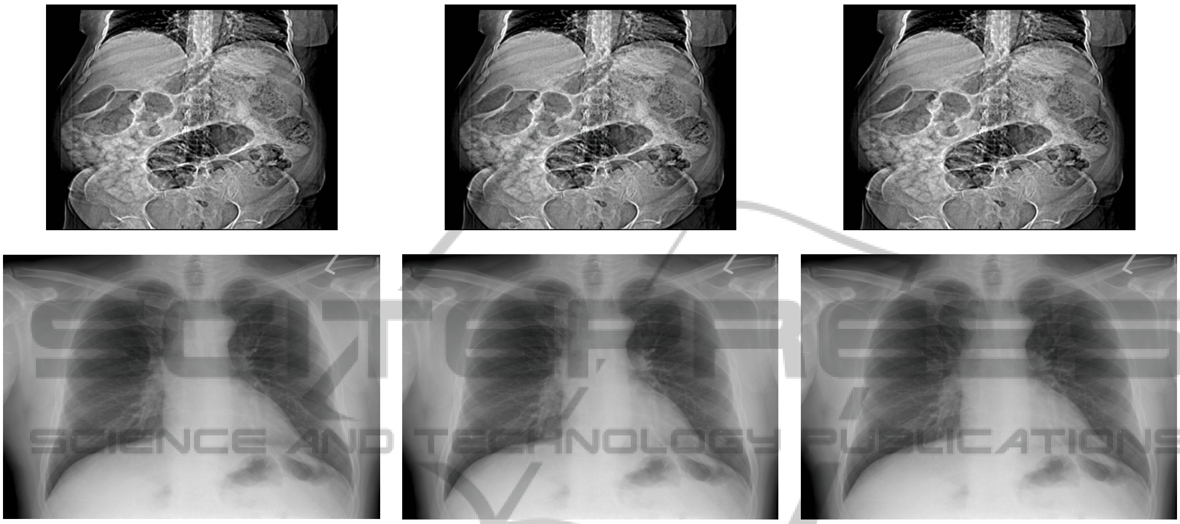

Figure 3: First and second row : original image, decompressed image , watermarked-decompressed image respectively.

From equations 8 and 11, it is clear that watermarking through *contexts* does not affect the prediction of pixels and hence the image quality.

Figure 3 gives the original image, decompressed image and watermarked decompressed image respectively.

From this figure 3 it is clear that the watermark does not effect the quality of the image as watermarking is performed on the *context* of the image. PSNR is given in Table I, and is computed as,

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(I(i,j) - I_w(i,j))^2 \tag{12}$$

$$PSNR = 10\log_{10}(255^2/MSE) \tag{13}$$

Mean square error (MSE), is the sum of squares of difference between the original image I and watermarked decompressed image I_w. The Structural Similarity Index (SSIM) measures the similarity between I and I_w and is given in Table I. SSIM is given as,

$$SSIM(I,I_w) = \frac{(2\mu_I\mu_{I_w} + c_1)(2\sigma_{II_w} + c_2)}{(\mu_I^2 + \mu_{I_w}^2 + c_1)(\sigma_I^2 + \sigma_{I_w}^2 + c_2)} \tag{14}$$

where, $\mu_{(.)}$ denotes mean, $\sigma_{(.)}$ denotes standard deviation, $c_1 = 6.5$ and $c_2 = 58.5$ are constants. PSNR and SSIM measures show that the decompressed image quality remains intact even after being watermarked.

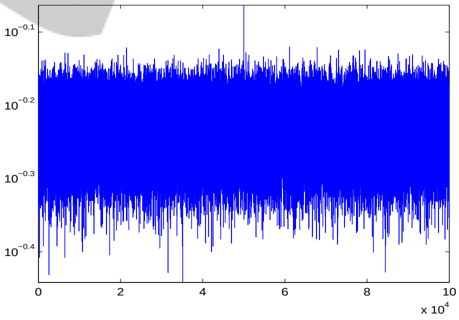

Figure 4: Detection against random watermarks (positive upward spike at center corresponds to embedded watermark)

4.3 Detection Performance

The detection performance is given in figure 4. For simulation 100000 watermarks are generated randomly and the average is reported for 10 randomly generated *context* set. It is evident from the figure 4 that the correct embedded watermark gives the highest correlation against the extracted watermark. On the experimental basis we find that a threshold $T > 0.8$ does not give any error in detection.

The watermark embedding takes place through *context* mapping. Since *context* values can only be exchanged with other *contexts*, attacks like additive noise, scaling, cropping, filtering or other attacks cannot be performed as these attacks will change the context value randomly further not rendering decoding.

The watermark can be attacked by changing the mapping f_w. Here for detection purpose, the watermark embedder can request the owner for the decryption key. Further decrypting the bit stream, the original *contexts* can be generated. Since the position of original *context* is known, the corresponding watermarked *context* position is also known. Thus the watermarked *context* can be retrieved and the detection follows similar to the process described in section 3.2. In this case the detection performance is similar as with the case of unattacked watermarked copy. This is because the absolute *context* value cannot be changed but only replaced with other *context*.

Another possible attack is collusion attack. In this case different watermarked copy holders may collude to extract or destroy the watermark. Colluders may get the highest frequency *contexts* and thus the corresponding watermarked *contexts*. However, this neither gives any information about the mapping used for watermarking nor the order in which *context* frequencies should be used (as the watermark is embedded using Euclidean distance criteria). Thus the attacker has to perform brute force attack to extract watermark.

4.4 Side Information

The side information i.e., the contexts and its frequencies occurring in original image is also given in Table I. It is clearly evident that the percentage of side information required is far less than the compressed filesize.

5 CONCLUSIONS

In this paper we propose a novel technique to embed a robust watermark in the JPEG-LS compressed and encrypted images. The algorithm is simple to implement as it is performed on the *context* of received compressed-encrypted media and does not require any decompression or decryption. Also the watermark is detected correctly in case of different attacks. The quality of the media is preserved as the watermark does not affect the pixel values itself, rather only the *context classmap* is changed. The side information required for watermarking is also very low as compared to the compressed filesize.

REFERENCES

Bianchi, T., Piva, A., and Barni, M. (2010). Composite signal representation for fast and storage-efficient processing of encrypted signals. *IEEE Transactions on Information Forensics and Security*, 5(1):180–187.

Blobel, B. (2004). Authorisation and access control for electronic health record systems. *International Journal of Medical Informatics*, 73(3):251–257.

Caldelli, R., Filippini, F., and Barni, M. (2006). Joint near-lossless compression and watermarking of still images for authentication and tamper localization. *Signal Processing: Image Communication*, 21(10):890–903.

Cancellaro, M., Battisti, F., Carli, M., Boato, G., De Natale, F., and Neri, A. (2008). A joint digital watermarking and encryption method. In *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X. Proc of the SPIE*, volume 6819, pages 68191C–68191C, 2008.

Fang, T. (2009). On performance of lossless compression for HDR image quantized in color space. *Signal Processing: Image Communication*, 24(5):397–404.

Katzenbeisser, S., Lemma, A., Celik, M., van der Veen, M., and Maas, M. (2008). A buyer–seller watermarking protocol based on secure embedding. *IEEE Transactions on Information Forensics and Security*, 3(4):783–786.

Lian, S., Liu, Z., Zhen, R., and Wang, H. (2006). Commutative watermarking and encryption for media data. *Optical Engineering*, 45:1–3.

Piva, A., Bianchi, T., and De Rosa, A. (2010). Secure client-side ST-DM watermark embedding. *Information Forensics and Security, IEEE Transactions on*, 5(1):13–26.

Schneier, B. (1996). *Applied Cryptography*. John Wiley and Sons, New York.

Thomas, T., Emmanuel, S., Subramanyam, A., and Kankanhalli, M. (2009). Joint watermarking scheme for multiparty multilevel DRM architecture. *IEEE Transactions on Information Forensics and Security*, 4(4):758–767.

Weinberger, M., Seroussi, G., and Sapiro, G. (2000). The LOCO-I lossless image compression algorithm: principles andstandardization into JPEG-LS. *IEEE Transactions on Image Processing*, 9(8):1309–1324.

Zhao, B., Kou, W., Li, H., Dang, L., and Zhang, J. (2010). Effective watermarking scheme in the encrypted domain for buyer-seller watermarking protocol. *Information Sciences*, pages 4672–4684.