ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

IJIREEICE

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

Designing an Efficient Image Encryption-then Compression System

B.Suneetha

Assistant Professor (ECE), Vardhaman College of Engineering ,Hyderabad, India[1]

Abstract: In many practical scenarios, image encryption has to be conducted prior to image compression. This has led to the problem of how to design a pair of image encryption and compression algorithms such that compressing the encrypted images can still be efficiently performed. In this paper, we design a highly efficient image encryption-then-compression (ETC) system, where both lossless and lossy compressions are considered. The proposed image encryption scheme operated in the prediction error domain is shown to be able to provide a reasonably high level of security. We also demonstrate that an arithmetic coding-based approach can be exploited to efficiently compress the encrypted images. More notably, the proposed compression approach applied to encrypted images is only slightly worse, in terms of compression efficiency, than the state-of-the-art lossless/lossy image coders, which take original, unencrypted images as inputs. In contrast, most of the existing ETC solutions induce significant penalty on the compression efficiency**.**

Keywords: Compression of encrypted image, encrypted domain signal processing, ETC.

I. INTRODUCTION

Consider an application scenario in which a content owner Alice wants to securely and efficiently transmit an image I to a recipient Bob, via an untrusted channel provider Charlie. Conventionally, this could be done as follows. Alice first compresses I into B, and then encrypts B into I_e using an encryption function $EK(*)$, where K denotes the secret key, as illustrated in Fig. 1(a). The encrypted data I_e is then passed to Charlie, who simply forwards it to Bob. Upon receiving I_e, Bob sequentially performs decryption and decompression to get a reconstructed image $\hat{I}$.

Even though the above *Compression-then-Encryption (CTE)* paradigm meets the requirements in many secure transmission scenarios, the order of applying the compression and encryption needs to be reversed in some other situations. As the content owner, Alice is always interested in protecting the privacy of the image data through encryption. Nevertheless, Alice has no incentive to compress her data, and hence, will not use her limited computational resources to run a compression algorithm before encrypting the data. This is especially true when Alice uses a resource-deprived mobile device. In contrast, the channel provider Charlie has an overriding interest in compressing all the network traffic so as to maximize the network utilization. It is therefore much desired if the compression task can be delegated by Charlie, who typically has abundant computational resources. A big challenge within such *Encryption-then-Compression (ETC)* framework is that compression has to be conducted in the encrypted domain, as Charlie does not access to the secret key K. This type of ETC system is demonstrated in Fig. 1(b). The possibility of processing encrypted signals directly in the encrypted domain has been receiving increasing attention in recent years [2]–[6]. At the first glance, it seems to be infeasible for Charlie to compress the encrypted data, since no signal structure can be exploited to enable a traditional compressor. Although counter-intuitive, Johnson *et. al* showed that the stream cipher encrypted data is compressible through the use of coding with side information principles, without compromising either the compression efficiency or the information-theoretic security [7]. In addition to the theoretical findings, [7] also proposed practical algorithms to losslessly compress the encrypted *binary* images. Schonberg *et. al* later investigated the problem of compressing encrypted images when the underlying source statistics is unknown and the sources have memory [8], [9]. By applying LDPC codes in various bit-planes and exploiting the inter/intra correlation, Lazzeretti and Barni presented several methods for lossless compression of encrypted grayscale/color images [11]. Furthermore, Kumar and Makur applied the approach of [7] to the prediction error domain and achieved better lossless compression performance on the encrypted grayscale/color images [12]. Aided by rate-compatible punctured turbo codes, Liu *et. al* developed a progressive method to losslessly compress stream cipher encrypted grayscale/color images [13]. More recently, Klinc *et al.* extended Johnson's framework to the case of compressing block cipher encrypted data [10].

To achieve higher compression ratios, lossy compression of encrypted data was also studied [14]–[20]. Zhang *et. Al* proposed a scalable lossy coding framework of encrypted images via a multi-resolution construction [14]. In [15], a compressive sensing (CS) mechanism was utilized to compress encrypted images resulted from linear encryption. A modified basis pursuit algorithm can then be applied to estimate the original image from the compressed and encrypted data. Another CS-based approach for encrypting compressed images was reported in [16]. Furthermore, Zhang designed an image encryption scheme via pixel-domain permutation, and demonstrated that the encrypted file can be efficiently compressed by discarding the excessively rough and fine information of coefficients in the transform domain [17]. Recently, Zhang *et. al* suggested a new compression approach for encrypted

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

images through multi-layer decomposition [18]. Extensions to blind compression of encrypted videos were developed in [19], [20].

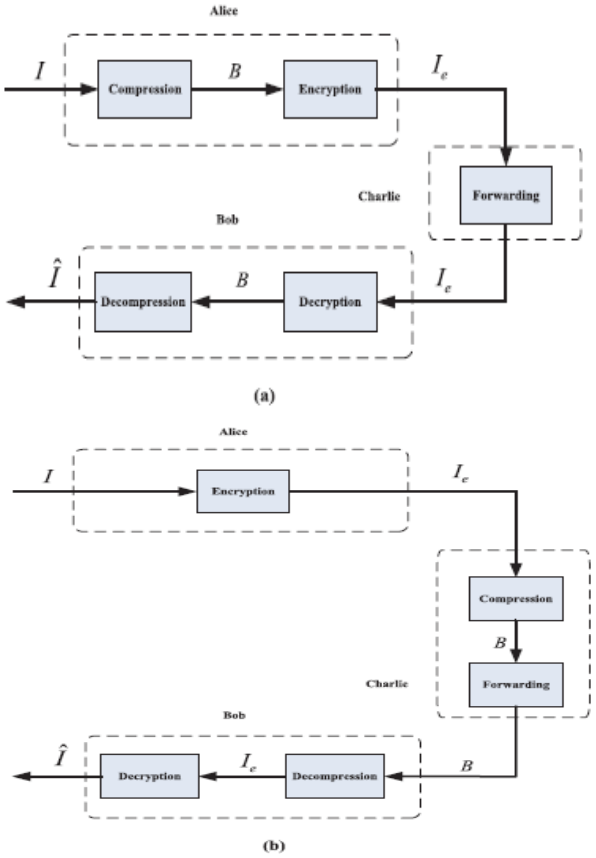

Fig. 1. (a) Traditional Compression-then-Encryption (CTE) system;
(b) Encryption-then-Compression (ETC) system.

Despite extensive efforts in recent years, the existing ETC systems still fall significantly short in the compression performance, compared with the state-of-the-art lossless/lossy image and video coders that require unencrypted inputs. The primary focus of this work is on the practical design of a pair of image encryption and compression schemes, in such a way that compressing the encrypted images is *almost* equally efficient as compressing their original, unencrypted counterparts. Meanwhile, reasonably high level of security needs to be ensured. If not otherwise specified, 8-bit grayscale images are assumed. Both lossless and lossy compression of encrypted images will be considered. Specifically, we propose a permutation-based image encryption approach conducted over the prediction error domain. A context-adaptive arithmetic coding (AC) is then shown to be able to efficiently compress the encrypted data. Thanks to the nearly i.i.d property of the prediction error sequence, negligible compression penalty (< 0.1% coding loss for lossless case) will be introduced. Furthermore, due to the high sensitivity of prediction error sequence against disturbances, reasonably high level of security could be retained.

The rest of this paper is organized as follows. Section II gives the details of our proposed ETC system, where lossless compression is considered. Extension to the case of lossy compression is given in Section III. In Section IV, we present the security analysis and evaluation of the compression performance. Experimental results are reported in Section V to validate our findings. We conclude in Section VI.

II. PROPOSED ETC SYSTEM

In this section, we present the details of the three key components in our proposed ETC system, namely, image encryption conducted by Alice, image compression conducted by Charlie, and the sequential decryption and decompression conducted by Bob.

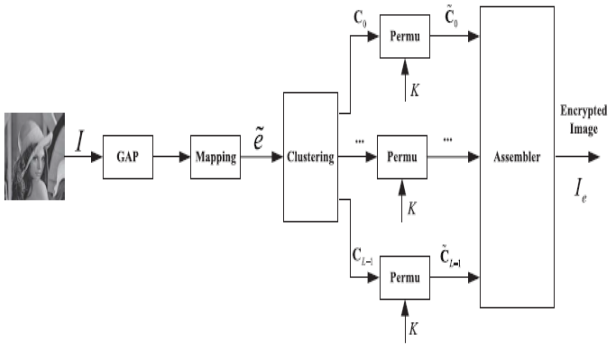

Fig. 2. Schematic diagram of image encryption

A. Image Encryption Via Prediction Error Clustering and Random Permutation:

From the perspective of the whole ETC system, the design of the encryption algorithm should simultaneously consider the security and the ease of compressing the encrypted data. To this end, we propose an image encryption scheme operated over the prediction error domain. The schematic diagram of this image encryption method is depicted in Fig. 2. For each pixel $I_{i,j}$ of the image I to be encrypted, a prediction $\bar{I}_{i,j}$ is first made by using an image predictor, e.g. GAP [21] or MED [22], according to its causal surroundings. In our work, the GAP is adopted due to its excellent de-correlation capability. The prediction result $\bar{I}_{i,j}$ can be further refined to $\tilde{I}_{i,j}$ through a context-adaptive, feedback mechanism [21]. Consequently, the prediction error associated with $I_{i,j}$ can be computed by

$$e_{i,j} = I_{i,j} - \tilde{I}_{i,j} \qquad (1)$$

Although for 8-bit images, the prediction error $e_{i,j}$ can potentially take any values in the range [−255, 255], it can be mapped into the range [0, 255], by considering the fact that the predicted value $\tilde{I}_{i,j}$ is available at the decoder side. From (1), we know that $e_{i,j}$ must fall into the interval $[-\tilde{I}_{i,j}, 255, -\tilde{I}_{i,j}]$, which only contains 256 distinct values. More specifically, if $\tilde{I}_{i,j} \leq 128$, we rearrange the possible prediction errors each of which is sequentially mapped to a value between 0 to 255. If $\tilde{I}_{i,j} > 128$, a similar mapping could be applied. Note that, in order to reverse the above mapping, the predicted value $\tilde{I}_{i,j}$ needs to be known. In the sequel, let us denote the mapped prediction error by $e_{i,j}$, which takes values in the range $[0, 255]$.

Our proposed image encryption algorithm is performed over the domain of the mapped prediction error $\hat{e}_{i,j}$. Instead of treating all the prediction errors as a whole, we

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

divide the prediction errors into L clusters based on a context-adaptive approach. The subsequent randomization and compression will be shown to be benefited from this clustering operation. To this end, an error energy estimator originally proposed in [21] is used as an indicator of the image local activities. More specifically, for each pixel location *(i, j)*, the error energy estimator is defined by

$$\Delta_{i,j} = d_h + d_v + 2|e_{i-1,j}| \quad (3$$

where

$$d_h = |I_{i-1,j} - I_{i-2,j}| + |I_{i,j-1} - I_{i-1,j-1}| + |I_{i,j-1} - I_{i+1,j-1}|$$
$$d_v = |I_{i-1,j} - I_{i-1,j-1}| + |I_{i,j-1} - I_{i,j-2}| + |I_{i+1,j-1} - I_{i+1,j-2}| \quad (3$$

and e_{i-1}, j is the prediction error at location $(i - 1, j)$. The design of the cluster should simultaneously consider the security and the ease of compressing the encrypted data. In an *off-line* training process, we collect a set of samples $(\hat{e}, \Delta)$ from appropriate training images. A dynamic programming technique can then be employed to get an optimal cluster in minimum entropy sense, i.e., choose $0 = q0 < q1 < \ldots < qL = \infty$ such that the following conditional entropy measure is minimized

$$\sum_{0 \le i \le L-1} H(\tilde{e}|q_i \le \Delta < q_{i+1}) p(q_i \le \Delta < q_{i+1})$$

Where $H(\cdot)$ is the 1-D entropy function taking logarithm in base 2. It can be seen that the term $H(\tilde{e}|q_i \le \Delta < q_{i+1})$ denotes the entropy of the prediction error sequence in the i th cluster, and hence, (4) becomes an approximation of the bit rate (in bpp) of representing all the prediction errors. Therefore, the cluster designed by minimizing (4) is expected to achieve optimal compression performance. Also, the selection of the parameter L needs to balance the security and the encryption complexity. Generally, larger L could potentially provide higher level of security because there are more possibilities for the attacker to figure out. However, it also incurs higher complexity of encryption. We heuristically find that $L = 16$ is an appropriate choice balancing the above two factors well. Note that the cluster configurations, i.e. the values of all q_i , are publicly accessible. For each pixel location *(i, j)*, the corresponding cluster index k can be determined by

$$k = \{k|q_k \le \Delta_{i,j} < q_{k+1}\} \quad (5$$

The algorithmic procedure of performing the image encryption is then given as follows:

Step 1: Compute all the mapped prediction errors $\hat{e}_{i,j}$ of the whole image I.

Step 2: Divide all the prediction errors into L clusters $\mathbf{C}_k$, for $0 \le k \le L - 1$, where k is determined by (5), and each $\mathbf{C}_k$ is formed by concatenating the mapped prediction errors in a raster-scan order.

Step 3: Reshape the prediction errors in each $\mathbf{C}_k$ into a 2-D block having four columns and $|\mathbf{C}_k|/4$ rows, where $|\mathbf{C}_k|$ denotes the number of prediction errors in $\mathbf{C}_k$.

Step 4: Perform two key-driven cyclical shift operations to each resulting prediction error block, and read out the data in raster-scan order to obtain the permuted cluster $\mathbf{C}_k$.

Let $\mathbf{CS}_k$ and $\mathbf{RS}_k$ be the secret key vectors controlling the column and the row shift offsets for $\mathbf{C}_k$. Here, $\mathbf{CS}_k$ and $\mathbf{RS}_k$ are obtained from the key stream generated by a stream cipher, which implies that the employed key vectors could be different, even for the same image encrypted at different sessions. The random permutation is also illustrated in Fig. 3 for an input sequence $S = s1s2 \ldots s16$, where the numbers within the blocks denote the indexes of the elements of S. Before permutation, the first row becomes *(*1, 2, 3, 4*)*, the second row becomes *(*5, 6, 7, 8*)*, etc. The column shifts are specified by a key vector **CS** = [2 3 0 1], with each column undergoing a downward cyclical shift in accordance with the key value associated with that column. The procedure is then repeated using another key vector **RS** = [1 3 1 2] for each of the rows. Note that such permutation operations can be realized via circular shifts, which are easily implemented in either hardware or software.

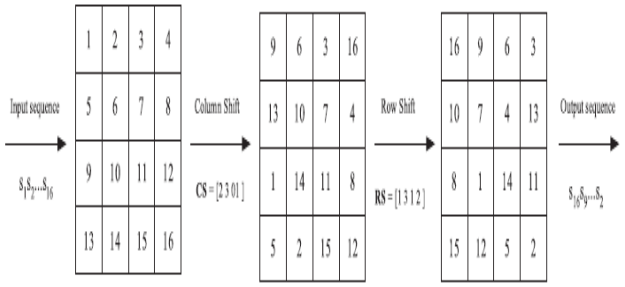

Fig. 3. An example of the cyclical shifts.

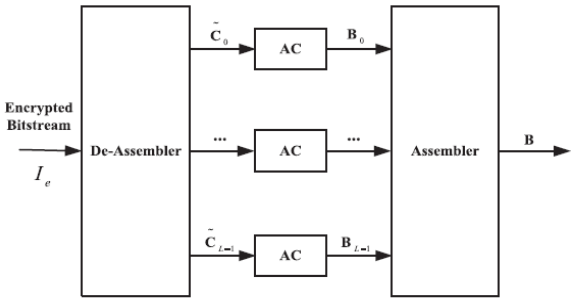

Fig. 4. Schematic diagram of compressing the encrypted data.

Step 5: The assembler concatenates all the permuted clusters $\mathbf{C}k$, for $0 \le k \le L - 1$, and generates the final encrypted image

$$I_e = \tilde{\mathbf{C}}_0 \tilde{\mathbf{C}}_1 \ldots \tilde{\mathbf{C}}_{L-1} \quad (6)$$

in which each prediction error is represented by 8 bits. As the number of prediction errors equals that of the pixels, the file size before and after the encryption preserves.

Step 6: Pass I_e to Charlie, together with the length of each cluster $|\mathbf{C}k|$, for $0 \le k \le L - 2$. The values of $|\mathbf{C}k|$ enable Charlie to divide I_e into L clusters correctly. In comparison with the file size of the encrypted data, the overhead induced by sending the length $|\mathbf{C}k|$ is negligible.

B. Lossless Compression of Encrypted Image Via Adaptive AC:

The compression of the encrypted file I_e needs to be performed in the encrypted domain, as Charlie does not

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

have access to the secret key K. In Fig. 4, we show the diagram of lossless compression of I_e. Assisted by the side information $|\tilde{\mathbf{C}}_k|$, for $0 \le k \le L-2$, a de-assembler can be utilized to parse I_e into L segments $\tilde{\mathbf{C}}_0, \tilde{\mathbf{C}}_1, \ldots, \tilde{\mathbf{C}}_{L-1}$ in the exactly same way as that done at the encryption stage. An adaptive AC is then employed to losslessly encode each prediction error sequence $\tilde{\mathbf{C}}_k$ into a binary bit stream $\mathbf{B}_k$. Note that the generation of all $\mathbf{B}_k$ can be carried out in a parallel manner to improve the throughput. Eventually, an assembler concatenates all $\mathbf{B}_k$ to produce the final compressed and encrypted bit stream $\mathbf{B}$, namely,

$$\mathbf{B} = \mathbf{B}_0\mathbf{B}_1 \ldots \mathbf{B}_{L-1} \quad (7)$$

Similar to the encryption stage, the length of $\mathbf{B}_k$, i.e. $|\mathbf{B}_k|$, for $0 \le k \le L-2$, needs to be sent to Bob as side information. The compressibility of each $\tilde{\mathbf{C}}_k$ relies on the fact that random permutation only changes the locations, while not the values of the prediction errors. This leads to the preservation of the probability mass function (PMF) of prediction error sequence, which drives the adaptive AC. The length of the resulting compressed bit stream can then be computed by

$$L_c = |\mathbf{B}| + (L-1)\lceil \log_2 |\mathbf{B}| \rceil \quad (8)$$

Where $|\mathbf{B}|$ is measured by bits, and the second term denotes the overhead induced by sending the side information $|\mathbf{B}_k|$, for $0 \le k \le L-2$.

Remark: In predictive coding such as the benchmark codec CALIC [21], over 50% of the computations come from the entropy coding part, assuming that the adaptive AC is adopted. This implies that if Alice has to compress the prediction errors via adaptive AC, the computational burden will at least be doubled.

C. Sequential Decryption and Decompression:

Upon receiving the compressed and encrypted bit stream $\mathbf{B}$, Bob aims to recover the original image I. The schematic diagram demonstrating the procedure of sequential decryption and decompression is provided in Fig. 5. According to the side information $|\mathbf{B}_k|$, Bob divides $\mathbf{B}$ into L segments $\mathbf{B}_k$, for $0 \le k \le L-1$, each of which is associated with a cluster of prediction errors. For each $\mathbf{B}_k$, an adaptive arithmetic decoding can be applied to obtain the corresponding permuted prediction error sequence $\tilde{\mathbf{C}}_k$. As Bob knows the secret key K, the corresponding de-permutation operation can be employed to get back the original $\mathbf{C}_k$.

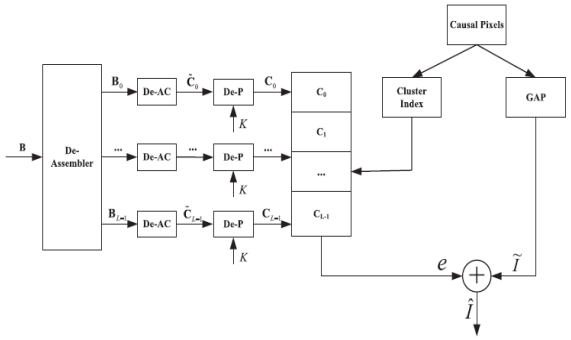

Fig. 5. Schematic diagram of sequential decryption and decompression.

With all the $\mathbf{C}_k$, the decoding of the pixel values can be performed in a raster-scan order. For each location (i, j), the associated error energy estimator $\Delta_{i,j}$ and the predicted value $I_{i,j}$ can be calculated from the causal surroundings that have already been decoded. Given $\Delta_{i,j}$, the corresponding cluster index k can be determined by using (5). The first *'unused'* prediction error in the kth cluster is selected as $\tilde{e}_{i,j}$, which will be used to derive $e_{i,j}$ according to $I_{i,j}$ and the mapping rule described in Section II-A. Afterwards, a '*used*' flag will be attached to the processed prediction error. The reconstructed pixel value can then be computed by

$$\hat{I}_{i,j} = \bar{I}_{i,j} + e_{i,j} \quad (9)$$

As the predicted value $I_{i,j}$ and the error energy estimator $\Delta_{i,j}$ are both based on the causal surroundings, the decoder can get the exactly same prediction $\tilde{I}_{i,j}$. In addition, in the case of lossless compression, no distortion occurs on the prediction error $e_{i,j}$, which implies $\hat{I}_{i,j} = I_{i,j}$, i.e., error-free decoding is achieved.

III. EXTENSION TO LOSSY COMPRESSION

In this section, we discuss the extension of our framework to provide lossy compression of encrypted images. A seemingly straightforward solution to this end is to let Charlie perform uniform scalar quantization on each element of $\mathbf{C}_k$, for $0 \le k \le L-1$, and then apply adaptive AC over quantized prediction errors. Unfortunately, this straightforward method leads to the undecodable problem, because the prediction $\tilde{I}_{i,j}$ is based on the original, unquantized surrounding pixels that are not available to the decoder side in the case of lossy compression.

To remedy this problem, quantization on prediction errors needs to be conducted by Alice. In other words, Alice has to be cooperative in order to gain the compression ratios. More specifically, after getting each prediction error $e_{i,j}$ via (1), Alice applies the following uniform scalar quantization on $e_{i,j}$ with a parameter τ

$$\check{e}_{i,j} = \begin{cases} (2\tau+1)\lfloor (e_{i,j}+\tau)/(2\tau+1) \rfloor & \text{if } e_{i,j} \ge 0 \\ (2\tau+1)\lfloor (e_{i,j}-\tau)/(2\tau+1) \rfloor & \text{if } e_{i,j} < 0 \end{cases} \quad (10)$$

where $\tilde{e}_{i,j}$ denotes the quantized version of $e_{i,j}$. Meanwhile, Alice maintains a reconstruction

$$\hat{I}_{i,j} = \bar{I}_{i,j} + \check{e}_{i,j} \quad (11)$$

which will be used to predict the subsequent pixels and establish the context models. In other words, the prediction and context modeling are now based on the causal reconstructed values $I_{i,j}$, rather than the original $I_{i,j}$. To achieve better compression performance, a similar mapping as that described in Section II-A will be conducted to narrow the range of $\check{e}_{i,j}$. For simplicity, we still use $\tilde{e}_{i,j}$ to represent the mapped version of $\check{e}_{i,j}$. In addition, the optimal cluster used to partition the error energy space needs to be re-designed in accordance to different τ. More specifically, for each τ, the training samples become $(\tilde{e}, \Delta)$, where $\tilde{e}$ is the mapped version of $\tilde{e}$ quantized with parameter τ and Δ is calculated with the reconstructed surrounding pixels. A dynamic programming technique can be similarly employed to get

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

the optimal cluster $0= q0(\tau) < q1(\tau) < \cdots < q_L(\tau)=\infty$, where the cluster configurations now depend on τ. As in the lossless case, all the values of $q0(\tau), q1(\tau), \ldots, q_L(\tau)$ are publicly accessible. The encrypted image is eventually constructed by concatenating the L clusters of quantized, permuted prediction error sequences, in a very similar fashion as that done in the lossless case.

Upon receiving the encrypted image, Charlie can retrieve the L clusters of quantized, permuted prediction errors. An adaptive AC can then be applied to encode the prediction errors in each cluster in a lossless way. Within the above framework of lossy compression of encrypted image, given fixed distortion, the lowest bit rate achievable R is determined by Alice through setting the quantization parameter τ. This is because the entropy of the prediction error sequences is fixed for given τ, which limits the lowest bit rate achievable. However, Charlie still enjoys the flexibility of adjusting the bit rate, which also depends on the compression algorithm applied, in addition to the parameter τ. For instance, Charlie may employ a non-adaptive Huffman coding to compress the prediction errors. Certainly, the resulting bit rate will be higher than that of the case when adaptive AC is used, while the complexity is lowered. In fact, Charlie may also select an even higher bit rate by compressing partial prediction errors and leaving the others in the uncompressed format, with even lowered complexity. Note that Charlie is not privileged to set a rate lower than R, because this results in lossy representation of the prediction error sequence. As will be discussed shortly in the next Section, tiny mismatch of the prediction error sequence still leads to severe degradation of the decoded image, causing it to be worthless. The ability of controlling the lowest achievable rate by the content owner may be treated as an advantageous feature of the proposed ETC scheme, since the quality of the decoded image at receiver side is guaranteed, though the manipulation of the encrypted data is completely andled by an untrusted party. In contrast, in many existing systems, e.g., [14] and [17], such guarantee cannot be offered, as Charlie can arbitrarily reduce the bit rate. Furthermore, in the case of lossy compression, the computational overhead at Alice's side will not be materially increased, as the uniform scalar quantization can be efficiently implemented. Noticing the fact that the dynamic programming operations for the optimal cluster design are performed completely off-line, the overall complexity of computation by Alice is not very high, and should be similar to that of some existing systems, e.g., [11] and [12], which are also operated over the prediction error domain.

IV. SECURITY AND PERFORMANCE ANALYSIS

In this section, we present the analysis regarding the security of the proposed permutation-based image encryption method and the efficiency of compressing the encrypted data.

A. Security Analysis

Recall that the key stream controlling the random permutation of $\mathbf{C}_k$ is generated by using a stream cipher. This implies that the key stream could be different even for the same image encrypted at different sessions. Hence, the only attack model applicable to our proposed encryption scheme is the ciphertext-only attack [23], in which the attacker can only access the ciphertext and attempts to recover the original image.

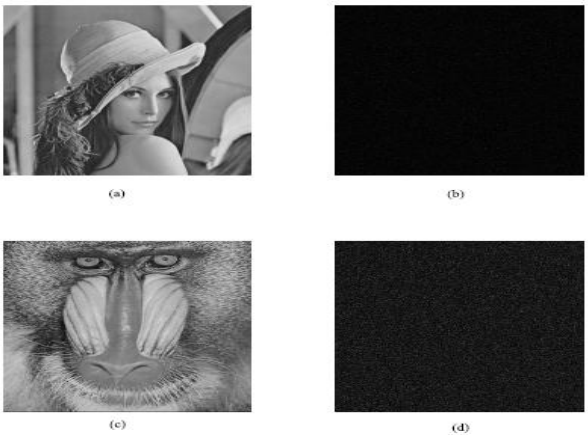

Fig. 6. (a) original Lena; (b) encrypted Lena; (c) original Baboon; (d)encrypted Baboon.

As the AC module is completely public and invertible, the attacker can obtain the encrypted image I_e, which is formed by concatenating L clusters of prediction error sequences. With I_e, the following statistical attack could be applied. As the length of each $\tilde{\mathbf{C}}_k$ is publicly known, I_e can be straightforwardly partitioned into L segments $\tilde{\mathbf{C}}_k$, for $0 \le k \le L-1$. For each $\tilde{\mathbf{C}}_k$, the empirical probability mass function (EPMF) can be calculated

$$p_k(i) = \frac{\#i}{|\tilde{\mathbf{C}}_k|} \qquad (12)$$

where $\#i$ denotes the number of i in $\tilde{\mathbf{C}}_k$ and $i \in [0, 255]$. The following conditional entropy quantity obtained by averaging
over L clusters can be used to measure the complexity of the input image

$$h = -\sum_{k=0}^{L-1} \frac{|\tilde{\mathbf{C}}_k|}{N} \sum_{i=0}^{255} p_k(i) \log_2 p_k(i) \qquad (13)$$

where N is the number of pixels in I. Clearly, images with intensive fine details would result in bigger values of h, while images with large portion of smooth regions would give smaller values. In other words, some statistical information of the input image leaks. However, it should be noted that statistical information leakage is inevitable for *any* feasible ETC systems. The feasibility of compressing the encrypted image without secret key allows any attacker to apply the same compression strategy on the encrypted data, and the size of the resulting file has already been a statistical indicator of the original image. For example, the Baboon image having more intensive statistical activities would lead to a larger file than Lena image.

Despite the statistical information leakage, it is practically intractable to figure out the permutation, due to the large number of distinct ways of performing the permutation. Specifically, the number of distinct ways of permutation for each $\mathbf{C}_k$ can be approximately calculated by

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

$$\frac{|\mathbf{C}_k|!}{\prod_{i=0}^{255}\left(\lfloor p_k(i)\cdot|\mathbf{C}_k|\rfloor!\right)} \qquad (14)$$

In practice, the number given by (14) is extremely large, precluding practical brute-force attack. For instance, the number of distinct ways of permuting the first cluster of Lena image is significantly larger than 2^{256}.

Fig. 7. Ten test images, each of which is assigned with an ID ranging from 1-10, in the raster-scan order. (a) Lena. (b) Peppers. (c) Goldhill. (d) Boat.(e) Man. (f) Harbor. (g) Airplane. (h) Barbara. (i) Bridge. (j) Tank.

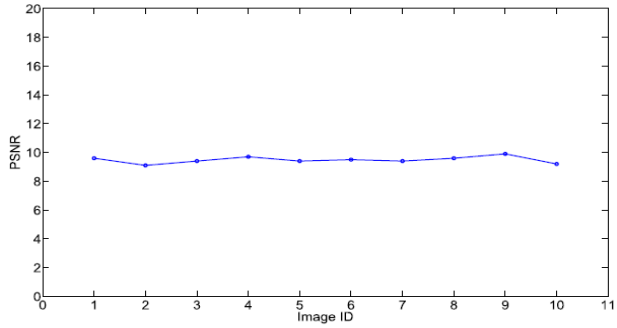

Fig. 8. Reconstruction results of directly decoding the compressed and encrypted images.

Alternatively, the attacker may attempt to decode the encrypted file I_e directly. For correct decoding of $I_{i,j}$, the attacker needs to get both the prediction error $\tilde{e}_{i,j}$ and the associated predicted value $\tilde{I}_{i,j}$. One way of guessing $\tilde{e}_{i,j}$ is to first estimate the cluster index k according to the decoded neighboring pixels, and then select one element from $\tilde{\mathbf{C}}_k$. Recall that the cluster index is determined according to the error energy estimator $\Delta_{i,j}$ calculated by using the casual surrounding pixels. This implies that the

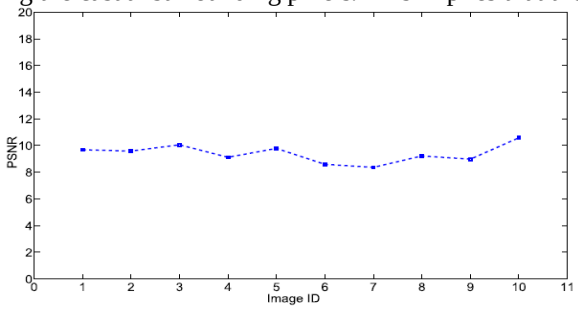

Fig. 9. Decoding performance of the case by assuming that bounded errors only occur in the prediction errors of the first two rows.

TABLE I
COMPARISON OF LOSSLESS COMPRESSION PERFORMANCE

Image	Proposed	CALIC	[13]	S_C	$S_{[13]}$
Lena	134267 B (4.096 bpp)	134232 B (4.096 bpp)	4.918 bpp	−0.03%	16.7%
Peppers	143998 B (4.394 bpp)	143974 B(4.394 bpp)	5.271 bpp	−0.02%	16.6%
Goldhill	150879 B(4.604 bpp)	150850 B(4.604 bpp)	5.453 bpp	−0.02%	15.6%
Boat	134732 B (4.112 bpp)	134651 B (4.109 bpp)	5.374 bpp	−0.06%	23.5%
Man	142404 B (4.346 bpp)	142361 B(4.345 bpp)	5.424 bpp	−0.03%	24.8%
Harbor	160552 B (4.900 bpp)	160487 B (4.898 bpp)	6.206 bpp	−0.04%	26.7%
Airplane	121261 B (3.701 bpp)	121172 B (3.698 bpp)	5.045 bpp	−0.07%	36.3%
Barbara	150352 B (4.588 bpp)	150223 B (4.584 bpp)	6.124 bpp	−0.09%	33.5%
Bridge	177311 B (5.411 bpp)	177252 B (5.409 bpp)	6.303 bpp	−0.03%	16.5%
Tank	154301 B (4.709 bpp)	154264 B (4.708 bpp)	5.370 bpp	−0.02%	14.0%
Averaged	4.452 bpp	4.449 bpp	5.575 bpp	−0.07%	25.3%

TABLE II
COMPARISON OF LOSSY COMPRESSION PERFORMANCE WHEN $\tau = 1$

Image	Proposed		CALIC		S_C
	rate	PSNR	rate	PSNR	
Lena	2.588	49.89	2.570	49.89	-0.70%
Peppers	2.868	49.89	2.848	49.89	-0.70%
Goldhill	3.074	49.89	3.051	49.89	-0.75%
Boat	2.626	49.90	2.601	49.90	-0.96%
Man	2.850	49.91	2.830	49.91	-0.70 %
Harbor	3.366	49.90	3.349	49.90	-0.51%
Airplane	2.302	49.91	2.277	49.91	-1.10 %
Barbara	3.074	49.89	3.056	49.89	-0.59 %
Bridge	3.890	49.91	3.868	49.91	-0.57 %
Tank	3.164	49.90	3.146	49.90	-0.57 %
Averaged	2.989	49.94	2.971	49.94	-0.61%

TABLE III
COMPARISON OF LOSSY COMPRESSION PERFORMANCE WHEN $\tau = 3$

Image	Proposed		CALIC		S_C
	rate	PSNR	rate	PSNR	
Lena	1.549	42.25	1.531	42.25	-1.18 %
Peppers	1.771	42.15	1.755	42.15	-0.91 %
Goldhill	1.969	42.20	1.950	42.2	-0.97 %
Boat	1.589	42.37	1.569	42.37	-1.27%
Man	1.814	42.33	1.797	42.33	-0.94 %
Harbor	2.260	42.20	2.243	42.20	-0.76%
Airplane	1.374	42.41	1.356	42.41	-1.33%
Barbara	2.025	42.20	2.010	42.20	-0.75%
Bridge	2.756	42.15	2.739	42.15	-0.62%
Tank	2.037	42.13	2.018	42.13	-0.94%
Averaged	1.960	42.28	1.941	42.28	-0.98%

TABLE IV
COMPARISON OF LOSSY COMPRESSION PERFORMANCE WHEN $\tau = 5$

Image	Proposed		CALIC		S_C
	rate	PSNR	rate	PSNR	
Lena	1.047	38.73	1.034	38.73	-1.26%
Peppers	1.239	38.40	1.223	38.40	-1.31 %
Goldhill	1.453	38.36	1.437	38.36	-1.11 %
Boat	1.132	38.84	1.115	38.84	-1.52%
Man	1.346	38.60	1.332	38.60	-1.04%
Harbor	1.694	38.54	1.677	38.54	-1.01 %
Airplane	0.931	38.80	0.904	38.80	-1.61%
Barbara	1.612	38.28	1.596	38.28	-1.00%
Bridge	2.180	38.22	2.163	38.22	-0.79%
Tank	1.508	38.24	1.492	38.24	-1.07%
Averaged	1.481	38.46	1.460	38.46	-1.44%

TABLE V
COMPARISON OF LOSSY COMPRESSION PERFORMANCE WHEN $\tau = 7$

Image	Proposed		CALIC		S_C
	rate	PSNR	rate	PSNR	
Lena	0.766	36.45	0.753	36.45	-1.73 %
Peppers	0.905	36.09	0.891	36.09	-1.57 %
Goldhill	1.137	35.85	1.121	35.85	-1.43%
Boat	0.875	36.41	0.861	36.41	-1.63%
Man	1.059	36.18	1.044	36.18	-1.42 %
Harbor	1.364	36.07	1.348	36.07	-1.19 %
Airplane	0.791	36.15	0.776	36.15	-1.93 %
Barbara	1.223	36.03	1.207	36.03	-1.33 %
Bridge	1.837	35.57	1.820	35.57	-0.93 %
Tank	1.162	35.71	1.148	35.71	-1.22 %
Averaged	1.185	35.90	1.164	35.90	-1.80%

casual pixels have to be correctly decoded, and any previous decoding error may cause error propagation, influencing the correct determination of the cluster index k. Furthermore, the accurate estimation of the predicted value $\tilde{I}_{i,j}$ also depends on the correct reconstruction of the casual surrounding pixels. In this sense, the error propagation effect inherent in predictive coding helps improve security. Even if k can somehow be correctly estimated, the attacker still needs to decide the value of $\tilde{e}_{i,j}$ within $\tilde{\mathbf{C}}_k$. As the distribution of the prediction errors is peaked at zero, the optimal estimation of $\tilde{e}_{i,j}$ in maximum likelihood (ML) sense is then zero. Unfortunately, setting all $\tilde{e}_{i,j} = 0$ does not lead to a semantically meaningful image. Another way of recovering the original image is to explicitly explore the spatial correlation of natural images when estimating the prediction errors. Suppose the attacker can somehow correctly estimate all the pixels up to $I_{i,j}$. Certainly, the prediction $\tilde{I}_{i,j+1}$ and the cluster index k for the pixel location $(i, j + 1)$ can be perfectly known, as all the casual surroundings are decoded without error. The attacker then needs to select one prediction error from the kth cluster. Due to the spatial correlation, if the selected $\tilde{e}_{i,j+1}$ makes the reconstruction $\hat{I}_{i,j+1}$ deviate too much from $I_{i,j}$, the attacker has high confidence to reject this selection. In other words, the spatial correlation helps the attacker narrow the set where the true prediction errors lie

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

in. To evaluate the feasibility and efficiency of this type of attack strategy, we actually consider a more general scenario, which is more advantageous for the attacker. It is now assumed that the attacker can perfectly estimate 99% of the prediction errors for the first two rows. We further assume that the estimation error is bounded by a small integer constant €, namely, $|\hat{e}_{i,j} - \tilde{e}_{i,j}| \leq €$, for all (i, j) in the first two rows, where $\hat{e}i, j$ denotes the estimation of $\tilde{e}_{i,j}$. All the other prediction errors in the remaining rows are assumed to be perfectly available. The above assumption, though impractical, offers us an opportunity to get an upper bound of the reconstruction performance that the attacker can achieve. As will be experimentally verified in the next Section, the PSNR values of the reconstructed images are still too low (around 10 dB), even under such favorable conditions. Therefore, the proposed permutation-based image encryption approach is still practically useful in those scenarios where perfect secrecy is not required.

B. Compression Performance

As a well-known fact, image predictors, e.g., GAP [21] and MED [22], have strong de-correlation capability, thus making the prediction error sequence nearly i.i.d. In other words, only small amount of inter-dependence exists in the prediction error sequence. Compared with the traditional predictive coding system in which the compression is conducted over the original, unpermuted prediction error sequences, the compression task of Charlie has to be performed over the *permuted* ones. From the perspective of information theory, a sequence with inter-dependence, which is caused by redundancy, is more compressible than its i.i.d counterpart. As the permutation operations in the prediction error domain destroy the inter-dependence, the resulting $\tilde{\mathbf{C}}_k$ is less compressible than its original, unpermuted version $\mathbf{C}_k$. Fortunately, the inter-dependence left in each $\mathbf{C}_k$ is rather limited, thanks to the superior de-correlation capability of image predictors. Hence, it is intuitively expected and will be verified by our experiments that the coding penalty caused by prediction error permutation is very small.

V. EXPERIMENTAL RESULTS

In this section, the security of our proposed image encryption and the compression performance on the encrypted data are evaluated experimentally. Fig. 6 illustrates the Lena and Baboon images, together with their encrypted versions, from which we can see that our encryption approach is effective in destroying the semantic meaning of the images. In addition, it can be observed that the encrypted Baboon image looks '*brighter*' than the encrypted Lena image. This is because the Baboon image contains large portion of texture regions that are difficult to compress, resulting in more large-valued prediction errors.

We implement the attack strategy of directly decoding the encrypted file I_e, as described in Section IV-A. Ten images of size 512 × 512 shown in Fig. 7 are used as the test set. In Fig. 8, we give the PSNR results of the reconstructed images, where x-axis represents the image ID. It can be observed that all the PSNR values are around 10 dB, which is too low to convey any useful semantic information. We also evaluate the reconstruction performance under the assumption that the bounded errors only occur in the prediction errors of the first two rows, while all the remaining ones are perfectly known. Here the estimation error bound ε is set to be 5. Fig. 9 illustrates the PSNR values of the reconstructed images, where each point is the averaged result of 10 realizations. It can be seen that, even under such favorable conditions, the attacker still cannot obtain any useful visual information of the source images, because all the PSNR values are too low (around 10 dB).

In Table I, the compression efficiency of our proposed method applied to the encrypted images is compared with the lossless rates given by the latest version of CALIC, a benchmark of practically good lossless image codecs, and the method in [13], a state-of-the-art lossless compression approach on encrypted images. The test set is composed of 100 images with various characteristics, 10 of which are shown in Fig. 7. Here 'B' and 'bpp' denote bytes and bit per pixel, respectively. In the rightmost two columns, *Sc* and *S*[13] stand for the bit rate saving of our proposed method over the CALIC and the method in [13], respectively. All the results of the proposed method are obtained by averaging over 10 random trials. The last row gives the averaged results over the 100 images in the test set. It can be seen that the coding penalty incurred by our method is consistently lower than 0.1% when comparing with the results of the CALIC. Meanwhile, the bit rate saving over the method in [13] can be up to 36.3%, which is achieved by the image Airplane. The lossy compression performance of the proposed method for different quantization parameters τ = 1, τ = 3, τ = 5, and τ = 7 is presented in Table II through V. The results of the near-lossless version of the CALIC (CALIC, in abbreviation) are also tabulated in these tables for comparison purpose. It can be observed that the coding penalty of our proposed method is less than 2%, compared with the CALIC. In addition, for fixed τ , both methods give the same PSNR values, as the distortion only depends on the prediction approach and the quantization strategy. We also notice that the coding penalty tends to increase for larger τ . Due to the employment of the uniform scalar quantizer, the l_∞ bound associated with the reconstructed image $\|I - \hat{I}\|_\infty = \tau$ still holds. This implies that further PSNR improvement can be retained by using our recently proposed soft decoding technique [24], at the cost of doubled _∞ bound. For simplicity, the results are omitted here.

In Fig. 10, we also compare the rate-PSNR performance of our compression method with JPEG 2000 and the method in [14]. For bit rates above 2 bpp, our method achieves even higher PSNR values than JPEG 2000. The gain in PSNR over JPEG 2000 can be significant for high bit rates. For instance, for the image Lena, the gain is more than 2 dB. As bit rate drops, the PSNR gain over JPEG 2000 decreases. When the bit rate is below 2 bpp, the PSNR gain over JPEG 2000 diminishes and starts to become negative. It can also be seen that the PSNR gain of our method over the one in [14] is quite remarkable. When the bit rate is around 2.50 bpp, the PSNR gain can be over

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

10 dB for the Lena image. We also notice that the method of [14] seems to suffer from the problem of performance saturation for images with intensive activities such as Harbor, Barbara, and Bridge.

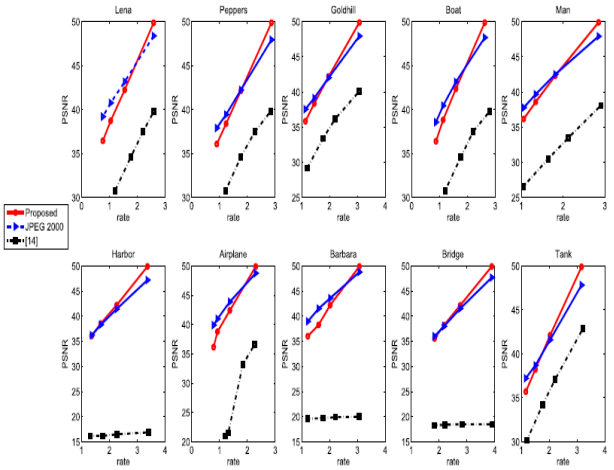

Fig. 10. Comparison of the rate-PSNR performance.

The rate-distortion behavior of our proposed scheme is mainly due to the nature of the coding structure, which essentially is a predictive coding. As a well-known fact, predictive coding always outperforms transform coding, not just JPEG 2000, as bit rate gets sufficiently high. This is because for high rate, bits will be allocated to code small high frequency coefficients in transform domain, and hence, the energy packing advantage of transform coding will be lost. When bit rate reduces, these small high frequency coefficients in transform domain will be quantized to zero, which can be very efficiently coded, making the energy packing advantage of transform coding apparent. For predictive coding, however, the quantizer is necessarily embedded in the prediction loop and this causes quantization errors to propagate. Error propagation occurs because future predictions are based on previously reconstructed values that are contaminated by quantization errors, resulting in prediction biases. Biased predictions can in turn generate even greater prediction errors, and so forth. The adverse effect of quantization error propagation becomes more complex if context-adaptive prediction is performed, in which case quantization errors would affect not only prediction but also context formation.

VI. CONCLUSION

In this paper, we have designed an efficient image Encryption-then-Compression (ETC) system. Within the proposed framework, the image encryption has been achieved via prediction error clustering and random permutation. Highly efficient compression of the encrypted data has then been realized by a context-adaptive arithmetic coding approach. Both theoretical and experimental results have shown that reasonably high level of security has been retained. More notably, the coding efficiency of our proposed compression method on encrypted images is very close to that of the state of-the-art lossless/lossy image codecs, which receive original, unencrypted images as inputs.

REFERENCES

[1] J. Zhou, X. Liu, and O. C. Au, "On the design of an efficient encryption then compression system," in Proc. ICASSP, 2013, pp. 2872–2876.
[2] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," IEEE Trans. Inf. Forensics Security, vol. 4, no. 1, pp. 86–97, Mar. 2009.
[3] T. Bianchi, A. Piva, and M. Barni, "Encrypted domain DCT based on homomorphic cryptosystems," EURASIP J. Inf. Security, 2009, Article ID 716357.
[4] 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.
[5] M. Barni, P. Failla, R. Lazzeretti, A.-R. Sadeghi, and T. Schneider, "Privacy-preserving ECG classification with branching programs and neural networks," IEEE Trans. Inf. Forensics Security, vol. 6, no. 2, pp. 452–468, Jun. 2011.
[6] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," IEEE Trans. Inf. Forensics Security, vol. 7, no. 3, pp. 1053–1066, Jun. 2012.
[7] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.
[8] D. Schonberg, S. C. Draper, and K. Ramchandran, "On blind compression of encrypted correlated data approaching the source entropy rate," in Proc. 43rd Annu. Allerton Conf., 2005, pp. 1–3.
[9] D. Schonberg, S. C. Draper, and K. Ramchandran, "On compression of encrypted images," in Proc. IEEE Int. Conf. Image Process., Oct. 2006, pp. 269–272.
[10] D. Klinc, C. Hazay, A. Jagmohan, H. Krawczyk, and T. Rabin, "On compression of data encrypted with block ciphers," IEEE Trans. Inf. Theory, vol. 58, no. 11, pp. 6989–7001, Nov. 2012.
[11] R. Lazzeretti and M. Barni, "Lossless compression of encrypted greylevel and color images," in Proc. 16th Eur. Signal Process. Conf., Aug. 2008, pp. 1–5.
[12] A. Kumar and A. Makur, "Distributed source coding based encryption and lossless compression of gray scale and color images," in Proc. MMSP, 2008, pp. 760–764.
[13] W. Liu, W. J. Zeng, L. Dong, and Q. M. Yao, "Efficient compression of encrypted grayscale images," IEEE Trans. Imag. Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.
[14] X. Zhang, G. Feng, Y. Ren, and Z. Qian, "Scalable coding of encrypted images," IEEE Trans. Imag. Process., vol. 21, no. 6, pp. 3108–3114, Jun. 2012.
[15] A. Kumar and A. Makur, "Lossy compression of encrypted image by compressing sensing technique," in Proc. IEEE Region 10 Conf. TENCON, Jan. 2009, pp. 1–6.
[16] X. Zhang, Y. L. Ren, G. R. Feng, and Z. X. Qian, "Compressing encrypted image using compressive sensing," in Proc. IEEE 7th IIHMSP, Oct. 2011, pp. 222–225.
[17] X. Zhang, "Lossy compression and iterative recobstruction for encrypted image," IEEE Trans. Inf. Forensics Security, vol. 6, no. 1, pp. 53–58, Mar. 2011.
[18] X. Zhang, G. Sun, L. Shen, and C. Qin, "Compression of encrypted images with multilayer decomposition," Multimed. Tools Appl., vol. 78, no. 3, pp. 1–13, Feb. 2013.
[19] D. Schonberg, S. C. Draper, C. Yeo, and K. Ramchandran, "Toward compression of encrypted images and video sequences," IEEE Trans. Inf. Forensics Security, vol. 3, no. 4, pp. 749–762, Dec. 2008.
[20] Q. M. Yao, W. J. Zeng, and W. Liu, "Multi-resolution based hybrid spatiotemporal compression of encrypted videos," in Proc. ICASSP, Apr. 2009, pp. 725–728.
[21] X. Wu and N. Memon, "Context-based, adaptive, lossless image codec," IEEE Trans. Commun., vol. 45, no. 4, pp. 437–444, Apr. 1997.
[22] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," IEEE Trans. Imag. Process., vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
[23] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Cleveland, OH, USA: CRC Press, 1997
[24] J. Zhou, X.Wu, and L. Zhang, "_2 restoration of _∞-decoded images via soft-decision estimation," IEEE Trans. Imag. Process., vol. 21, no. 12, pp. 4797–4807, Dec. 2012.

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 3, Issue 3, March 2015

BIOGRAPHY

B.SUNEETHA Post Graduated in Systems and Signal Processing M.Tech From JNTU, Hyderabad in 2011 and Graduated in Electronics and Communications Engineering B.Tech from G.PULLA REDDY ENGINEERING COLLEGE, KURNOOL, in 1996. She is working as Assistant Professor in Department of Electronics and Communications Engineering in VARDHAMAN COLLEGE OF ENGINEERING, R.R Dist, A.P, India. She has 12+ years of Teaching Experience. Her Research Interests in VLSI and COMMUNICATIONS.