International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Improving the Image Encryption-Then-Compression System via Prediction Error Cluster and Random Transformation

[1]N.Priyanka & [2]S.Md.Yousuf

[1]M, Tech (Embedded system), SVEW, Tirupati , Email Id: priyankanelluru@gmail.com
[2]Assistant professor, Dept of ECE, SVEW, Tirupati, Email Id: yousuf.digital@gmail.com

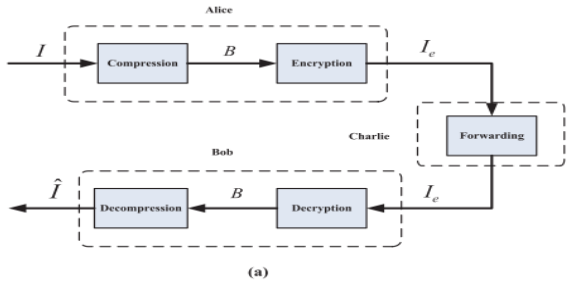
Abstract—

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 project, design a highly efficient image encryption-then-compression (ETC) system, where both lossless and lossy compression are considered. Image encryption scheme operated over the prediction error domain. In gradient adaptive prediction (GAP) is adopted due to its excellent de-correlation capability. Divide the prediction errors into clusters based on a context-adaptive approach. It perform cyclical shift operations to each resulting prediction error block and read out the data in raster-scan order to obtain the permuted cluster and generates the final encrypted image. 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. Lossless Compression design an arithmetic coding (AC) based approach to efficiently compress the encrypted image. It can be shown that the proposed scheme can provide reasonably high level of security and efficiency. To measure compressed efficiency of proposed method to the encrypted images is compared with lossless rates given by Context Based Adaptive Lossless Image Codec (CALIC).In lossy compression of encrypted image, perform uniform scalar quantization on each element and then apply adaptive arithmetic coding(AC) over quantized prediction errors. It varies quantization parameters to measure peak signal to noise ratio and mean square error. It is high level of security and efficient.

Index Terms: Compression of encrypted image; encrypted domain signal processing.

1. 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 Ie using an encryption function E_K (.), where K denotes the secret key, as illustrated in Fig. 1(a). The encrypted data Ie is then passed to Charlie, who simply forwards it to Bob. Upon receiving Ie, Bob sequentially performs decryption and decompression to get a reconstructed image Î. 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



(a)

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

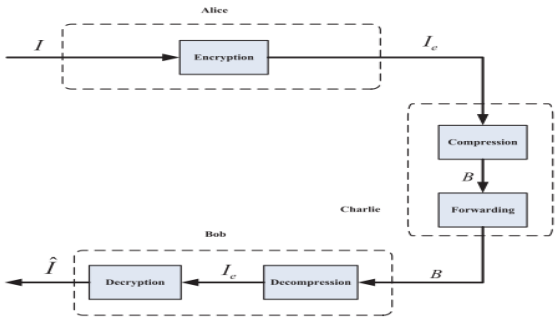


Fig. 1. (a) Traditional Compression-then-Encryption (CTE) system;(b)Encryption-then-Compression(ETC)system.

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. Furthermore, Kumar and Makur applied the approach to the prediction error domain and achieved better lossless compression performance on the encrypted grayscale/color images [11]. Aided by rate-compatible punctured turbo codes, Liu et. al developed a progressive method to losslessly compress stream cipher encrypted grayscale/color images. More recently, Klinc et al. extended Johnson's framework to the case of compressing block cipher encrypted data .To achieve higher compression ratios, lossy compression of encrypted data was also studied . Zhang et. al proposed a scalable lossy coding framework of encrypted images via a multi-resolution construction. In 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 [12]. 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. Recently, Zhang et. al suggested a new compression approach for encrypted images through multi-layer decomposition. Extensions to blind compression of encrypted videos were developed. 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 project is organized as follows. Gives the details of proposed ETC system, where lossless compression is considered. Extension to the case of lossy compression presents the security analysis and evaluation of

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

the compression performance. Experimental results are reported in to validate our findings.

II. RELATED WORK

In the last years, we have witnessed the coincidence of several key factors, such as the popularization of social networks and the creation of multiple web services that store and process personal data in environments out of the control of the data owner. This fact raised the issue of personal data privacy, therefore questioning the legality and morality of the use of such data by untrustworthy parties. Furthermore, European data protection directives require a high level of privacy protection for personal and sensitive data in virtually any context.

The emergent discipline of Signal Processing in the Encrypted Domain (SPED), born as a result of the joint efforts of the cryptographic community and signal processing community, provides efficient technological means to enforce privacy protection in signal processing applications. This target is achieved through the development of malleable encryption schemes and secure protocols for sensitive data and signals that allow for the execution of operations directly on the encrypted signals, with no access to them in the clear. Hence, the application of SPED techniques preserves users' privacy even when their data are stored and processed in an untrusted environment, like a public Cloud. It has an active research line in SPED, with a broad theoretical-practical scope, that has been materialized in recent years in numerous publications and contributions to international journals and conferences, and several international patient applications in the area of secure signal processing. The privacy models and primitives developed within the cryptographic concepts like Secure Multiparty Computation and Secure Function Evaluation. However, we have not left aside the practical approach given by the numerous applications of this technology, some of them being

- Protection of biometric signals in access control systems
- Secure adaptive filtering
- Privacy protection in outsourced multimedia Clouds
- Privacy protection in videosurveillance systems
- Privacy in fine-grained Smart Metering applications
- Data mining on private databases
- Secure applications for eHealth (telediagnosis /telemedicine, e.g., DNA analysis)
- Traceability of copyright infringements through private insertion of watermarks

III. 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.

Image Encryption Via Prediction Error Clustering and Random Permutation:

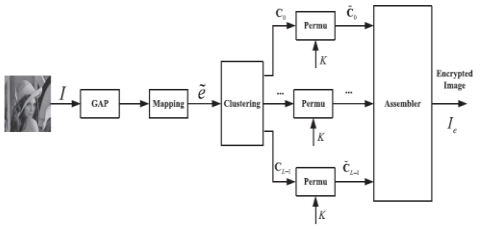


Figure:2 Schematic diagram of image encryption.

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 *Ii, j* of the image *I* to be encrypted, a prediction Ii, *j* is first made by using an image predictor, e.g. GAP or MED, according to its causal surroundings. In our work, the GAP is adopted due to its excellent de-correlation capability. The prediction result Ii, *j* can be further refined to ˜ *Ii, j* through a context-adaptive, feedback mechanism. Consequently, the prediction error associated with *Ii, 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 *ei, 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 ˜ *Ii, j* is available at the decoder side. From (1), we know that *ei, j* must fall into the interval [− ˜ *Ii, j* , 255− ˜ *Ii, j*], which only contains 256 distinct values. More specifically, if ˜ *Ii, j* ≤ 128, we rearrange the possible prediction errors. − ˜ *Ii, j* ,− ˜ *Ii, j* + 1, . . . , 0, 1, . . . , ˜ *Ii, j* , ˜ *Ii, j* + 1, . . . , 255 − ˜ *Ii, j* in the order 0,+1,−1, . . . ,+ ˜ *Ii, j* ,− ˜ *Ii, j* , ˜ *Ii, j* + 1, ˜ *Ii, j* + 2, . . . , 255 − ˜ *Ii, j* , each of which is sequentially mapped to a value between 0 to 255. If ˜ *Ii, j* > 128, a similar mapping could be applied.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Note that, in order to reverse the above mapping, the predicted value ˜ *Ii, j* needs to be known. In the sequel, let us denote the mapped prediction error by ˜*ei, j* , which takes values in the range [0, 255]. Our proposed image encryption algorithm is performed over the domain of the mapped prediction error ˜*ei, j* . Instead of treating all the prediction errors as a whole, we 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

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

Where

$$d_h = |I_{i-1,j} - I_{i-2,j}| + |I_{i,j-1} - I_{i-1,j-1}| + |I_{i,j-1} - I_{i+1,J+1}|$$

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

and *ei*−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 *(˜e,_)* 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 = *q*0 < *q*1 < · · · < *qL* = ∞ such that the following conditional entropy measure is minimized

$$\sum_{0\le i\le L-1} H(e\sim\backslash q_i \le \Delta < q_{i+1}) p(q_i \le \Delta < q_{i+1}) \qquad (4)$$

where *H(·)* is the 1-D entropy function taking logarithm in base 2. It can be seen that the term *H(˜e|qi* ≤ _ < *qi*+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 *qi* , are publicly accessible. For each pixel location *(i, j)*, the corresponding cluster index *k* can be
determined by

$$k = \{k\backslash q_k \le \Delta_{i,j} < q_{k+1} \qquad (5)$$

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

Step 1: Compute all the mapped prediction errors ˜*ei, j* of the whole image *I* .

Step 2: Divide all the prediction errors into *L* clusters **C***k* , for 0 ≤ *k* ≤ *L* − 1, where *k* is determined by (5), and each **C***k* is formed by concatenating the mapped prediction errors in a raster-scan order.

Step 3: Reshape the prediction errors in each **C***k* into a 2-D block having four columns and _|**C***k* |/4_ rows, where |**C***k*| denotes the number of prediction errors in **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 ˜**C***k* .

Let **CS***k* and **RS***k* be the secret key vectors controlling the column and the row shift offsets for **C***k* . Here, **CS***k* and **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* = *s*1*s*2 . . . *s*16, 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.

Step 5: The assembler concatenates all the permuted clusters ˜**C** *k*, for 0 ≤ *k* ≤ *L* − 1, and generates the final encrypted image *Ie* = ˜**C**0 ˜**C** 1 . . . ˜**C***L*−1 (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.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

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

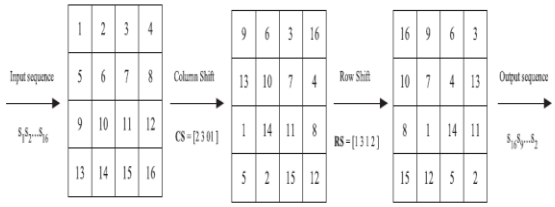


Figure:3 An example of the cyclical shifts.

4.1.2Lossless Compression of Encrypted Image Via Adaptive AC:

The compression of the encrypted file *Ie* needs to be performed in the encrypted domain, as Charlie does not have access to the secret key *K*. In Fig. 4, we show the diagram of lossless compression of *Ie*. Assisted by the side information $|\tilde{\mathbf{C}}k|$, for $0 \le k \le L-2$, a de-assembler can be utilized to parse *Ie* 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 **B**, namely, $\mathbf{B} = \mathbf{B}0\mathbf{B}1 \ldots \mathbf{B}L-1$ (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

$$L_c = |B| + (L-1)[\log_2 |B|] \qquad (8)$$

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

Sequential Decryption and Decompression:

Upon receiving the compressed and encrypted bit stream **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 **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 $\mathbf{C}k$.

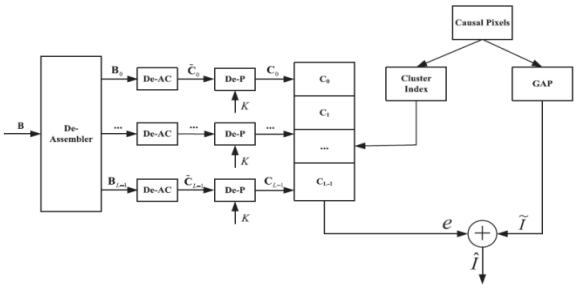


Fig. 4. Schematic diagram of sequential decryption and decompression

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 *_i, j* and the predicted value *˜ Ii, j* can be calculated from the causal surroundings that have already been decoded. Given *_i, j* , the corresponding cluster index *k* can be determined by using (5). The first *'unused'* prediction error in the *k*th cluster is selected as *˜ei, j*, which will be used to derive *ei, j* according to *˜ Ii, 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

$$\hat{I}_{i,j} = \tilde{I}_{i,j} + e_{i,j}$$

As the predicted value *˜ Ii, j* and the error energy estimator *_i, j* are both based on the causal surroundings, the decoder can get the exactly same prediction *˜ Ii, j* . In addition, in the case of lossless compression, no distortion occurs on the prediction error *ei, j* , which implies *ˆ Ii, j = Ii, j* , i.e., error-free decoding is achieved.

IV. EXTENSION TO LOSSY COMPRESSION:

The extension of our framework to provide lossy compression of encrypted images. A seemingly straight forward solution to this end is to let Charlie perform uniform scalar quantization on each element of $\tilde{\mathbf{C}}k$, for $0 \le$

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

$k \leq L - 1$, and then apply adaptive AC over quantized prediction errors. Unfortunately, this straightforward method leads to the un decodable problem, because the prediction ˜ *Ii, j* is based on the original, un quantized 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 *ei, j* via (1), Alice applies the following uniform scalar quantization on *ei, j* with a parameter τ

$$e_{i,j} = \begin{cases} (2\tau + 1\ (e_{i,j} + \tau)/(2\tau + 1)) \\ (2\tau + 1\ (e_{i,j} - \tau)/(2\tau + 1)) \end{cases}$$

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

where ˇ*ei, j* denotes the quantized version of *ei, j* . Meanwhile, Alice maintains a reconstruction

$$\hat{I}_{i,j} + \tilde{I}_{i,j} + \check{e}_{i,j}$$

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 ˆ *Ii, j* , rather than the original *Ii, j* . To achieve better compression performance. A will be conducted to narrow the range of ˇ*ei, j* . For simplicity, we still use ˜*ei, j* to represent the mapped version of ˇ*ei, 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 *(˜e,_)*, where ˜*e* is the mapped version of ˇ*e* quantized with parameter τ and _ is calculated with the reconstructed surrounding pixels. A dynamic programming technique can be similarly employed to get the optimal cluster 0= *q*0*(τ)*<

*q*1*(τ)* < · · · < *qL(τ)*=∞, where the cluster configurations now depend on τ . As in the lossless case, all the values of *q*0*(τ), q*1*(τ),* . . . *, qL(τ)* 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 handled by an un trusted party. In contrast, in many existing systems, 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, which are also operated over the prediction error domain.

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

EXPERIMENTAL RESULTS

Figure 5: Input image & Error Image

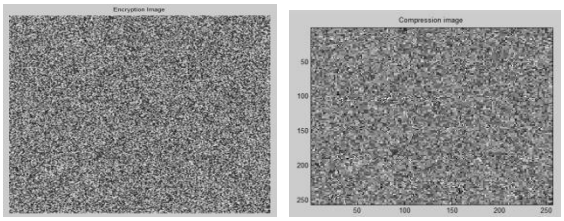


Figure 6: Encrypted &compression

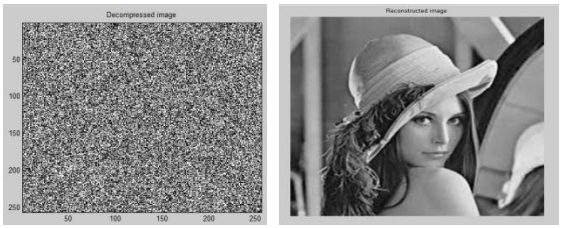


Figure 7: Decryption & Lossless compressed image

Figure 8: Lossy compressed Image

PERFORMANCE EVALUATION OF PROPOSED SYSTEM

Table:1 Lossless compression bit rate

S.No	Image	Proposed System	Existing system
1	Lena	3.816(Bpp)	4.096(Bpp)

Table:2 Lossy Compression parameters of Lena Image

S.No	T values	Proposed System		Existing system	
		PSNR	RATE	PSNR	RATE
1	t=1	60.418	3.220	49.89	2.570
2	t=3	57.238	2.311	42.25	1.531
3	t=5	54.059	1.789	38.73	1.034
4	t=7	50.879	3.816	36.45	0.753

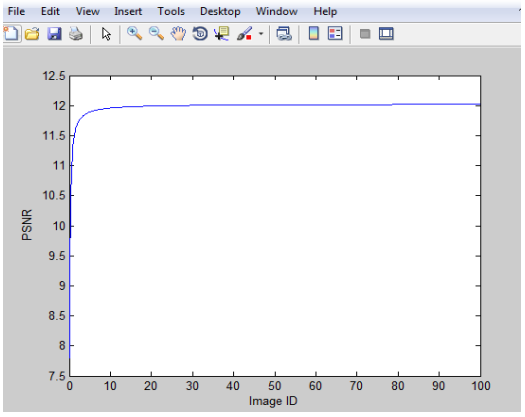


Figure 9: Average performance PSNR & Image ID

International Journal of Research
International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

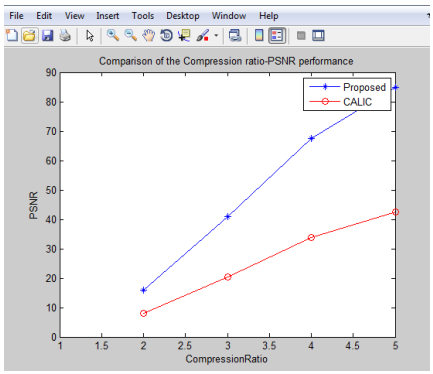


Figure 10: Average performances of PSNR & Compression ratio

IV.CONCLUSION

In this project, 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 stateof- the-art lossless/lossy image codecs, which receive original, unencrypted images as inputs.

REFERENCES

[1] Prabhakaran et al, _On Compressing Encrypted Data_, IEEE Trans on Signal Proc Vol 52 No. 10, pp2992-3006 Oct 2004

[2] T. Bianchi, A. Piva, and M. Barni, "Encrypted domain DCT based on homomorphic cryptosystems," *EURASIP J. Inf. Security*, 2009, Article ID 716357.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] S. S. Pradhan and K. Ramchandran, Distributed source coding us- ing syndromes (DISCUS): Design and construction,_ IEEE Trans. In- form.Theory, Vol. 49, pp. 626_643, Mar. 2003.

[10] D. Slepian and J. K. Wolf, _Noiseless coding of correlated information sources,_ IEEE Trans. Inform. Theory, Vol. IT-19, pp. 471_480, July 1973.

[11] A.Wyner and J. Ziv, _The rate-distortion function for source coding with side information at the decoder,_ IEEE Trans. Inform. Theory, Vol. IT-22, pp. 1_10, Jan. 1976.

[12] 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.

[1]**N.Priyanka,** her pursuing M.Tech, Embedded systems, Sri Venkateswara Engineering College for Women, Tirupati, priyankanelluru@gmail.com

[2]**S.Md.Yousuf** his obtain B.E in Anna University, Chennai and M.Tech, Jntu, Anantapur, Present his working in Sri Venkateswara Engineering College for Women Tirupati. **yousuf.digital@gmail.com**