Systematic Intensification in Encryption and Compression of Natural Image Data using Prediction, Clustering and Permutation

Sudhindra Sathyanarayana.
Information Science,
AMC Engineering College.

Pavithra N.
Assistant Professor,
AMC Engineering College.

***Abstract*—As most practical scenarios demand, image compression should be performed after encryption of an image. But carrying out compression of an already encrypted image has to be performed with lot of efficiency. Whereas, this has led to the trouble of creating enhanced encryption and compression algorithms for image data. Under the current thesis, an intensified systematic encryption-then-compression method is developed, considering both lossless compressions. The recommend encryption method for natural image data, operated under prediction error domain is said to contribute to fairly high level of reliability. Moreover, arithmetic coding-based approach has been beneficial in efficiently compressing an encrypted data of an natural image. Particularly, when it comes to compression efficiency of image data, the recommended approach of compression when applied over encrypted image data is worse, by just a few measures than the state-of-the-art lossless techniques which need actual original image data or the unencrypted image data for an input. Many existing Encryption then Compression methodologies have a significant influence on compression efficiency. This influence has caused certain amount of penalty, which has been overcome under here, and achieved a noticeably better performance.**

Keywords—Compression of encrypted image, Image encryption, Encrypted domain signal processing, Permutation.

I. INTRODUCTION

Let us consider a scenario of application in which a data content owner Alice wants to efficiently and securely transmit an image data I to particular recipient Bob, via an mistrusted provider of channel Charlie. This could be done as follows. Alice first performs image compression of image I into B, and then performs encryption of B into I_e by using an encryption function $E_K(\cdot)$, 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 $\hat{I}$.

Despite the above paradigm which is Compression-then-Encryption, which meets the requirements in several secure transmission scenarios, the order of applying the compression and then 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 so she ensure it by performing image encryption. Yet, Alice has no enough resources to compress her data, and hence, will not use her limited computational resources to run a compression algorithm before performing the encrypting on the image data. It is possible that when Alice uses a resource-deprived tablet or an mobile device. As opposed to this, the channel provider Charlie is keen in compressing all the traffic in going through his network so as to maximize the network utilization. Hence it is highly desired if the compression task can be delegated by Charlie, who basically has loads of computational resources. Thus now, the question within such Encryption-then-Compression framework is that operation of compression has to be conducted within encrypted domain, as Charlie cannot have access to the secret key K.

The above described variety of Encryption-then-Compression system is depicted in Fig. 1(b). The chances of processing any encrypted signals directly under the encrypted domain, have been receiving increasing attention in recent years [2]–[6]. Initially, it seems to be not feasible for network handler Charlie to compress the encrypted image data, since no signal structure can be utilized to enable a traditional compressor. Albeit counter-intuitive, Johnson et. al have showed that the encryption data whose encryption is done by stream cipher method, is compressible through the use of coding with side information principles, without compromising the efficiency in compression or the security related to the information-theoretic [7]. Besides the theoretical findings, [7] have further proposed algorithms which are practical and can perform lossless compression, of the encrypted binary image data. Schonberg et. al further investigated the trouble of compressing encrypted images when the underlying source statistics are not known and the data sending sources have manageable memory [8], [9]. Lazzeretti and Barni have presented various methods for lossless compression of particularly encrypted grayscale/color images [11] by employing LDPC codes in various bit-planes and exploiting the inter/intra correlation. Also, Kumar and Makur have applied the approach used by [7] upon prediction error domain and have achieved better performance on lossless compression with the encrypted grayscale/color images [12]. Liu *et. al* have developed a method which is progressive and which performs lossless compression on the grayscale images using the stream cipher compress technique on the stream cipher encryption[13], when assisted by rate-compatible punctured turbo codes. Recently though, Klinc *et al.* have extended Johnson's framework to the case of compressing block cipher encrypted data [10].

I. Related Work

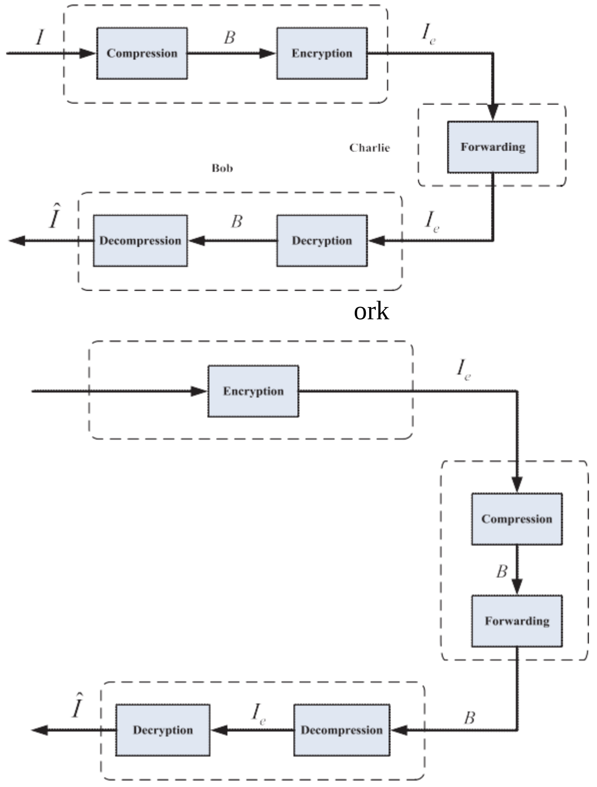

Fig. 1. (a) Traditional Compression-then-Encryption system;(b) Encryption-then-Compression system

In [1], Demijan Klinc et al, examines compression of data encrypted with block ciphers, as an example, the Advanced Encryption Standard. It is demonstrated that such data can be practically compacted without information of the mystery key. Block ciphers working in different binding modes are considered and it is demonstrated how compression can be attained to without bargaining security of the encryption plan. Further, it is demonstrated that there exists an essential constraint to the commonsense compressibility of block ciphers when no tying is utilized between blocks. Some execution results for reasonable code developments used to pack parallel sources are displayed. In[2], Ricardo Lazzeretti
et al, proposed possibility of lossless pressure of encrypted images has been as of late exhibited by depending on the relationship with source coding with side data at the decoder. However past works just tended to the pressure of bilevel images, specifically scanty high contrast images, with uneven probabilities of highly contrasting pixels. In this paper we examine the likelihood of compressing encrypted dark level and shading images, by decaying them into bit-planes. A couple ways to deal with adventure the spatial and cross-plane correlation among pixels are examined, and the likelihood of misusing the correlation between shading groups. Some exploratory results are demonstrated to assess the hole between the proposed arrangements and the hypothetically achievable execution. In [3], Mauro Barni et al, demonstrated privacy protection is an essential issue in numerous biomedical signal processing applications. Hence, specific consideration has been given to the utilization of secure multiparty computation procedures for processing biomedical signals, whereby nontrusted gatherings have the capacity to control the signals despite the fact that they are encrypted. This paper concentrates on the advancement of a privacy saving programmed analysis framework whereby a remote server characterizes a biomedical signal gave by the customer without getting any data about the signal itself and the last aftereffect of the characterization. In particular, we present and think about two techniques for the protected order of electrocardiogram (ECG) signals: the previous taking into account straight fanning projects (a specific sort of choice tree) and the last depending on neural systems.

II. PROPOSED SYSTEM

As the substance proprietor, Alice is constantly keen on securing the protection of the picture information through encryption. In any case, Alice has no motivator to compress her information. Interestingly, the channel supplier Charlie has an overriding enthusiasm for compressing all the network movement in order to augment the network utilization. It is in this manner quite sought if the compression assignment can be appointed by Charlie, who commonly has bottomless computational assets.

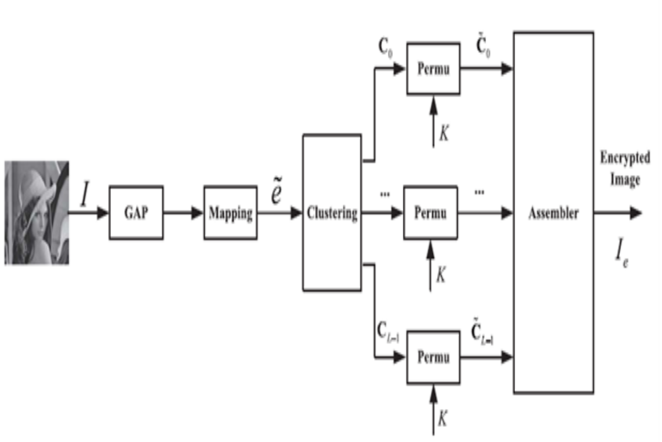

Fig 2. System Architecture

An enormous test inside such Encryption-then-Compression (ETC) structure is that compression must be directed in the encoded area, as Charlie does not access to the mystery key K. Here we can ascertain the anticipated error focuses utilizing a system known as GAP. There will be a head pixel which will progressively give us the quantity of error expectation pixels furthermore relying upon the extent of every pixel we can figure the quantity of cluster that we can structure inside a specific picture and the undertaking of the head pixel is that it will gather the closest n number of pixel that we oblige and structure a cluster. Consequently the time needed for the arrangement of the cluster will be less.

The algorithmic technique of performing the picture encryption is then given as takes after: Process all the mapped forecast errors, of the entire picture. Then all the forecast errors are partitioned into clusters and every cluster is framed by linking the mapped expectation errors in a raster-sweep request. Reshape the forecast errors in every cluster into a 2-D block having four sections and lines, where every cluster indicates the quantity of expectation errors in every clusters.

Perform two key-driven cyclical movement operations to every subsequent forecast error block, and read out the information in raster-output request to acquire the permuted cluster. Let segment moves and column movements be the mystery key vectors controlling the section and the line movement balances for every cluster. Here, segment moves and column movements are gotten from the key stream produced by a stream figure, which suggests that the utilized key vectors could be distinctive, actually for the same picture encoded at diverse sessions.

The arbitrary stage is likewise represented, for an info succession, where the numbers inside the blocks mean the records of the components of every arrangement. Before stage, the first column gets to be as (1, 2, 3, 4), the second line turns into (5, 6, 7, 8), and so on. The section movements are indicated by a key vector, with every segment experiencing a descending cyclical move as per the key worth connected with that section. The technique is then rehashed utilizing another key vector for line shifts. for each of the columns. Note that such stage operations can be acknowledged through round movements, which are effectively executed in either equipment or programming.

The constructing agent connects all the permuted clusters, and creates the last scrambled picture, in which every expectation error is spoken to by 8 bits. As the quantity of forecast errors squares with that of the pixels, the record estimate previously, then after the fact the encryption jam. Pass the new picture through the transmission channel, at the flip side, together with the length of every cluster. Then the estimations of empower the transmission channel to partition the picture into number of clusters accurately. In examination with the record size of the encoded information, the overhead impelled by sending the length is insignificant.

A. *PREPROCESSING*

The image is loaded and converted into the entropy format and the gradient pixel is identified. Image pre-processing is the term for operations on images at the lowest level of abstraction. These operations do not increase image information content but they decrease it if entropy is an information measure. The aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis task. Image pre-processing use the redundancy in images. Neighboring pixels corresponding to one real object have the same or similar brightness value. If a distorted pixel can be picked out from the image, it can be restored as an average value of neighboring pixels. Image pre-processing methods can be classified into categories according to the size of the pixel neighborhood that is used for the calculation of new pixel brightness. Image pre-processing can significantly increase the reliability of an optical inspection. Several filter operations which intensify or reduce certain image details enable an easier or faster evaluation. Users are able to optimize a camera image with just a few clicks.

B. *GRADIENT ADJUSTED PREDICTOR (GAP)*

Gradient Adjusted Predictor is a simple, adaptive nonlinear predictor that ca adapt itself to the intensity gradients near the predicted pixel. Hence it is more robust than the traditional DPCM like linear predictors, particularly in areas of strong edges. GAP differs from the existing linear predictors in that it weights the neighboring pixels of $I_{i,\ j}$ according to the estimated gradients of the image. In GAP the gradient of the intensity function at the current pixel I is estimated by computing the respective quantities. This is further explained as follows. 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, 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. 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 $\tilde{I}$ i, j is available at the decoder side. From (1), we know that ei, 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 $-\tilde{I}_{i,j}, -\tilde{I}_{i,j} + 1, \ldots, 0, 1 \ldots \tilde{I}_{i,j}, \tilde{I}_{i,j} + 1, \ldots, 255 - \tilde{I}_{i,j}$ in the order $0, +1, -1, \ldots, +\tilde{I}_{i,j}, -\tilde{I}_{i,j}$, $\tilde{I}_{i,j} + 1, \tilde{I}_{i,j} + 2, \ldots, 255 - \tilde{I}_{i,j}$, 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 $\tilde{e}_{i,j}$ which takes values in the range [0, 255].

Our proposed image encryption algorithm is performed over the domain of the mapped prediction error $\tilde{e}_{i,j}$. 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 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 ($\tilde{e}$, Δ) 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 < • • • < q L = ∞ such that the following conditional entropy measure is minimized

$$\sum_{0\le i\le L-1} H(\tilde{e} \;|\, q_i \le \Delta < q_{i+1}) p(q_i \le \Delta < q_{i+1}) \qquad (4)$$

where $H(\bullet)$ 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 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

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

C. MAPPING & CLUSTERING

Compute all the mapped prediction errors $\tilde{e}_{i,j}$ of the whole image *I*. Divide all the prediction errors into *L* clusters **C**$_k$, for $0 \le k \le L - 1$, where *k* is determined by (5), and each **C**$_k$ is formed by concatenating the mapped prediction errors in a raster-scan order. Reshape the prediction errors in each **C**$_k$ into a 2-D block having four columns and $\lceil |C_k|/4 \rceil$ rows, where $|C_k|$ denotes the number of prediction errors in **C**$_k$.

D. PERMUTATION

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 $\tilde{C}_k$. 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.

E. ASSEMBLER

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

$$I_e = \tilde{C}_0\, \tilde{C}_1 \,.\,.\,.\, \tilde{C}_{L-1} \qquad (6)$$

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. Pass *Ie* to Charlie, together with the length of each cluster $|\tilde{C}_k|$, for $0 \le k \le L - 2$. The values of $|\tilde{C}_k|$ 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 $|\tilde{C}_k|$ is negligible.

III. RESULTS

CONCLUSION

As per this present thesis, an efficient image Encryption-then-Compression 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 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 encryptionthen-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.