ISSN : 0976-8491(Online) | ISSN : 2229-4333(Print)

Compression Technique for Encrypted Images

[1]G.Hemanth Kumar, [2]P.Ravi Kiran, [3]V.Satyanarayana, [4]Dr. A.Yesubabu
[1,2,3,4]Dept. of CSE, Swarnandhra College o f Engineering & Technology, Narsapur, A.P., India

Abstract
Lossless compression of encrypted sources can be achieved through Slepian-Wolf coding. For encrypted real-world sources such as images, the key to improve the compression efficiency is how the source dependency is exploited. Approaches in the literature that make use of Markov properties in the Slepian-Wolf decoder do not work well for grayscale images. In this paper, we propose a resolution progressive compression scheme which compresses an encrypted image progressively in resolution, such that the decoder can observe a low-resolution version of the image, study local statistics based on it, and use the statistics to decode the next resolution level. Good performance is observed both theoretically and experimentally.

Keywords
compression of encrypted images, Slepian-Wolf coding, resolution progressive compression.

I. Introduction

Conventionally in secure transmission of redundant data, as illustrated in fig. 1(a), the data is usually first compressed and then encrypted at the sender side; to recover the data at the receiver side, decryption is performed prior to decompression. However, in some application scenarios, this conventional diagram needs to be revisited. Let us consider the following case (fig. 1(b)). Suppose Alice needs to send information to Bob, while Charlie is the network provider. Alice wants to keep the information confidential to Charlie, however the resources that she has is too limited to perform compression. So Alice just encrypts the data using a simple cipher and gets it forwarded. Charlie, as the network provider, always has the interest to reduce the data rate.

A practical system to compress encrypted data is also proposed. For example, suppose the plaintext X is an Independent and Identically Distributed (I.I.D.) source, and Alice uses a stream cipher (e.g., RC4 or DES in CFB mode) as the encryption function to generate the cipher text Y:

$$Y = X \oplus K \quad (1)$$

Where⊕ denotes the bit-wise exclusive OR operation, and K is the key stream.

Charlie gets Y without knowing K. He will encode Y through random binning: each sequence of n samples (denoted in Y^n) is randomly thrown into one of the bins that are indexed as $\{1, 2... 2^{nR}\}$. Only the bin indices are transmitted (which means the actual sending rate for Y is R). Bob,upon receiving the bin indices as well as the secret key, will treat the key stream as the side information and look for its joint typical pical sequence Yˆn inside the given bin. According to the Slepian-Wolf theorem [4], if and only if

$$R \geq H(Y|K) = H(X \oplus K|K) = H(X), \quad (2)$$

the reconstruction of Y^n can be asymptotically error-free. Finally the plaintext is reconstructed as

$$\hat{X} = \hat{Y} \oplus K. \quad (3)$$

Equation(2) basically suggests that theoretically, by employing Slepian-Wolf coding, the compression efficiency of the cipher text can be just as good as compressing the plaintext. On the other hand, very good practical Slepian-Wolf codes have been found recently that can approach the theoretical bound for ideal sources; trellis coded vector quantization and Slepian-Wolf coding are used for lossy compression of encrypted I.I.D. Gaussian sources.

However, challenges still remain when it comes to practical applications. Considering real-world sources such as images or videos, which are typically highly correlated, a critical issue in improving the coding efficiency is how to exploit the source dependency. Conventional encoder-side decor relation methods such as transform or prediction are not applicable here because the encryption function has masked the source dependency. In the literature, solutions have been proposed to treat image and video data as Markov sources and to exploit the Markovian property in the Slepian-Wolf decoder. A similar work is also found for non-encrypted colored sources. Some good results have been reported for binary images. However, there are some limitations with this approach: first, Markov decoding in a Slepian-Wolf decoder is expensive, especially in dealing with sources with non-binary alphabets; second, bit-plane based Markov decoding certainly reduces the complexity, but the source dependency that originally defined in the symbol domain is usually not fully utilized when translated to bit-planes; third, since image and video data are known to be highly non-stationary, a global Markov model cannot describe its local statistics precisely. As reported for 8-bit grayscale images, only the first 2 most significant bit-planes (MSB) are compressible by employing a 2-D Markov model in bit-planes. To the best knowledge of the authors, how to effectively exploit the correlation of encrypted image data remains a challenging issue.

In this paper, we propose an efficient way to compress encrypted images through resolution-progressive compression (RPC). The encoder starts by sending a down sampled version of the cipher text. At the decoder, the corresponding low-resolution image is decoded and decrypted, from which a higher-resolution image is obtained by intra-frame prediction. The predicted image, together with the secret encryption key, is used as the side information (SI) to decode the next resolution level. This process is iterated until the whole image is decoded. By doing so, the task of de-correlating the pixels, which is not possible for the encoder, is shifted to the decoder side. In addition, by having access to a lower-resolution image, the decoder is able to learn the local statistics, doing much better than “blind” decoding. Moreover, by avoiding exploiting the Markovian property in Slepian-Wolf decoding, the decoder’s complexity is significantly reduced.

II. Resolution Progressive Compression Of Encrypted Images

A. System Description

The encoder gets the cipher text Y and decomposes it into four sub-images, namely, the 00, 01, 10 and 11 sub-images. Each sub-image is a down sampled-by-two version of the encrypted image. The name of a sub-image denotes the horizontal and vertical offsets of the down sampling. The 00 sub-image is further down sampled to create multiple resolution levels. We use 00_n to represent the 00 sub-image in the n^{th} resolution level. The 00_n sub-image can be synthesized from the 00_{n+1}, 01_{n+1},

10_{n+1} and 11_{n+1} sub-images. An example of the decomposition is illustrated in fig. 2 Here the image is supposed to be an encrypted one. We show it in plaintext just for a better illustration. Meanwhile, we would like to point out that the stream cipher function in (1) only scrambles the pixel values, but does not shuffle the pixel locations. This means geometric information of the pixels is still preserved, which is leveraged by the down sampling operation.

After the down sampling, each sub-image is encoded independently using Slepian-Wolf codes, and the resulting syndrome bits are transmitted from the lowest resolution to the highest. Decoding starts from the 00 sub-image of the lowest-resolution level, say, level N. We suggest transmitting

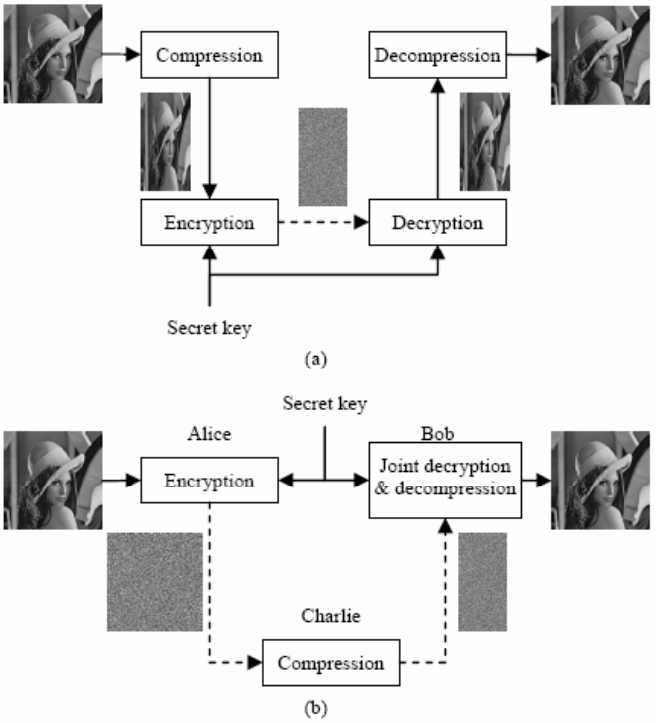

Fig. 1: Secure transmission of redundant data using (a) the conventional approach in which data is first compressed and then encrypted and (b) compression of encrupted data. Here soid arrows represent secure channels and dashed arrows denoted public channels

the uncompressed 00N sub-image as the doped bits. Thus the 00N sub-image can be known by the decoder without ambiguity, and knowledge about the local statistics will be derived based on it. Next, other sub-images of the same resolution level are interpolated from the decrypted 00N sub-image. We call the interpolation result the SI of the plaintext. It will then be scrambled the same way as in (1) to generate the SI of the cipher text. Since it is a one-to-one mapping between SI of the plaintext and SI of the cipher text, for the sake of clarity, we use "SI" only for the former in the rest of the paper. Meanwhile, a channel estimation module is employed to estimate the conditional Probability Density Function (P.D.F.) of the original pixel values, given the SI. The SI, the estimated P.D.F., and the corresponding part of the key stream are fed into the Slepian-Wolf decoding module to decode the target sub-image (Fig. 3 shows an example). When the 01_N, 10_N and 11_N sub-images are all decoded and decrypted, the 00_{N-1} sub-image can be synthesized, then the decoding iterates until the full-resolution image is reconstructed. It is worth noting that if the SI is a good approximation of the target sub-image, the pixels in the target sub-image can be considered as conditionally independent of each other (given the SI). In this case it is not necessary for the Slepian-Wolf decoder to exploit the Markovian property of the source, which greatly reduces the computational complexity.

A feedback channel is needed for the encoder to know how many bits to transmit for each sub-image, which generally increases the transmission delay. However, this cost is reasonable because the encoder has no idea about the source statistics and cannot determine the coding rate. It is the decoder who is able to learn such information and advise the encoder. On the other hand, the feedback channel does consume some bandwidth, but the consumption is not directly related to the compression efficiency, and the amount of information transmitted through the feedback channel is minimal.

B. Context Adaptive Interpolation

The SI generation in our scheme is through interpolation. For the sake of simplicity, for any pixel in the target sub-image, we only use the 4 horizontal and vertical neighbors or the 4 diagonal neighbors in the known sub-image(s) for the interpolation. Intuitively, the SI quality will be better, if the neighbors are geometrically closer to the pixel to be interpolated. Hence we use a two-step interpolation in each resolution level to improve the SI estimation. First, sub-image 11 is interpolated from sub-image 00; after sub-image 11 is decoded, we use both 00 and 11 to interpolate 01 and 10.

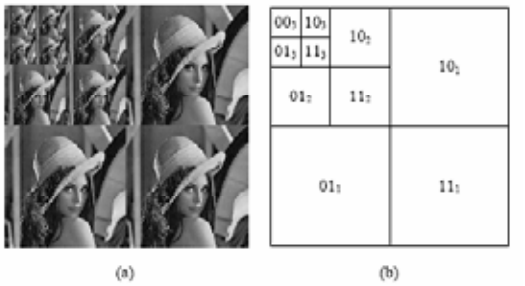

Fig. 2 : (a) Illustration of a three-level decomposition of the unencrypted image and (b) layout of the sub-images.

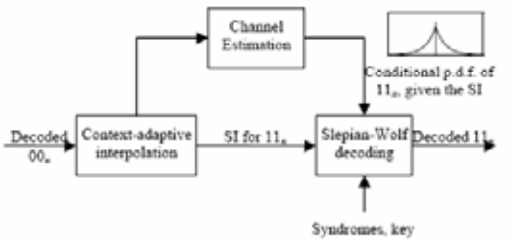

Fig. 3: Decoder's diagram in decoding the 11_n sub image

The interpolation pattern is illustrated in fig. 4, from which we can see another benefit of the two-step interpolation: the interpolation patterns of the two steps are isomorphic up to a scaling factor of $\sqrt{2}$ and a rotation of $\pi/4$. This simplifies the interpolator design.

Real-world image data is highly non-stationary; hence it is desired to have the interpolation adapted to the local context. For example, for a pixel on an edge, it is preferable to interpolate along the edge orientation. Similar efforts can be found in conventional lossless image compression, where the median edge detector (MED) [14] and the radiant adaptive predictor (GAP) [15] are two successful context adaptive predictors. However, they process the pixels in a raster-scanning order,

thus cannot be directly applied to our scheme.

In this subsection, a simple, yet effective context adaptive interpolator (CAI) is proposed for our scheme. Due to the isomorphism, we only describe the horizontal-vertical interpolator illustrated in fig. 4

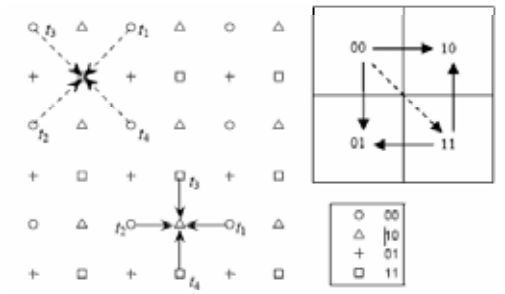

Fig. 4: Illustration of two-step interpolation at the decoder side. The dashed arrows denote the first step interpolation, and the solid arrows denote the second step.

Let s be the pixel value to be interpolated, $t = [t_1, t_2, t_3, t_4]^T$ be the vector of neighboring pixels. The interpolator classifies the local region into four types: smooth, horizontally-edged, vertically-edged and other. In smooth regions, a mean filter is applied; in horizontally/vertically edged regions, the interpolation is done along the edge; otherwise we use a median filter. More specifically, the proposed CAI is formulated as in (4) it can be verified that the first condition contradicts both the second and the third conditions, thus a "smooth" region will never be estimated as "edged" again. The second and the third conditions are adapted from GAP, with an ad hoc threshold. It is also possible that the region is diagonally-edged, but there is no clue about on which side of

$$s \begin{cases} mean(t) & (\max(t) - \min(t) \le 20) \\ (t_1 + t_2)/2 & (|t_3 - t_4| - |t_1 - t_2| > 20) \\ (t_3 + t_4)/2 & (|t_1 - t_2| - |t_3 - t_4| > 20) \\ Median(t) & (Otherwise) \end{cases} \quad (4)$$

the edge s lies. Therefore we simply adopt a median filter in this case. One might be concerned about error propagation due to the prediction structure: if one pixel is decoded in error in a lower resolution level, will the error propagate to higher-resolution levels? In fact, Slepian-Wolf coding has good error resiliency. Even if some errors occur in a lower resolution, it only makes the SI worse and increases the conditional entropy based on such SI. In this case, we might need more syndrome bits to make the Slepian-Wolf decoding converge, but as long as it converges, the error probability vanishes.

C. Localized Channel Estimation

Slepian-Wolf decoding treats the SI as a noisy version of the source to be decoded. We can consider that there is a virtual channel between the source and the SI [10]. To perform Slepian-Wolf decoding, it is also necessary for the decoder to estimate the statistics of the virtual channel. In this work, we adopt similar settings as in [16] and model the conditional P.D.F. of a pixel s to be Laplacian, centered at the given SI ˆs

$$P(s \mid \hat{s}) = \alpha \frac{\alpha}{2} \exp(\alpha |s - \hat{s}|) \quad (5)$$

Where $\alpha = \sqrt{\frac{2}{\delta^2}}$, and is the variance of P(s | ŝ). Hence it is necessary for the channel to estimate δ^2

Fig. 5: Illustration of the localized channel estimation. The conditional variance of a pixel in the $1\,0_n$ sub-image is referred to its neighbors in the $1\,0_{n-1}$ sub-image.

Due to the non-stationary of image data, the accuracy of the SI could vary a lot in different areas. Generally σ2s\ sˆ at smooth areas will be much smaller than that at textured areas. Therefore it is desirable to have a localized channel estimator. In this work, σ2s\ sˆ is estimated from the neighboring prediction (interpolation) residual of the previously decoded level. As illustrated in fig. 5, Let s be a pixel in the 10n sub-image, the channel estimator observes several geometrical neighbors of s in the 10_{n+1} sub-image. The neighborhood is chosen to be a 5×5 window. The mean square error (MSE) of the CAI results for these pixels is scaled to be used as σ2s\ sˆ . The scaling is needed because for interpolation at higher-resolution levels, the correlation between the neighboring pixels is higher, which usually means smaller prediction residual. In this work, we adopt an empirical scaling factor of 0.75.

It can be seen that, both the CAI and the localized channel estimation are based on the assumption that the decoder has access to a lower-resolution reconstruction of the image. In other words, they are both enabled by the resolution-progressive decoding.

D. Performance Analysis

Theoretical analysis will be provided for RPC. Compared to the state-of-the-art conventional lossless image coding schemes such as JPEG-LS and CALIC, RPC may suffer from two types of rate loss. The first type is the source coding loss, which is caused by the inefficiency of channel codes in achieving the Shannon limit. The second type is the image coding loss, caused by the inefficiency in removing the redundancy among the pixels. In this subsection we focus on the second type of loss. We will use an ideal source model to analyze the performance gap. Although real image data are not that simple, the analysis will still provide useful insights and help us understand the performance limit of RPC.

Moreover, the result helps the design of our channel estimating module. If the wide-sense stationary model is assumed, the scaling factor for estimating should be $\sigma 2_{s\backslash \hat{s}} \cdot F(2_n)/F(2_n+)2$ for the 11_n sub-image, or be F(2n-1)/F(2n+)2 for the 10_n and 11_n sub-images. Numerical results show that F(k)/F(k+2) typically ranges from 0.5 to 1, depending on the ρ value. So, we adopt 0.75 as the scaling factor. More sophisticated modeling might further improve the coding performance.

III. Simulation Results

The images listed in Table I are used for testing. The encoder decomposes each encrypted image into 4 resolution levels. The sub-images in the lowest-resolution level are sent without compression. But the decoder still performs inter sub-image interpolation. The results will be used to estimate the conditional P.D.F. of the pixels in the next level. For the other sub-images, we transmit the four least significant bit-planes (LSB) as raw bits, because there is not much gain to employ Slepian-Wolf coding on them. The four LSBs are sent prior to the MSBs, such that the decoder can have better knowledge about the pixels before starting decoding the MSBs. The four MSBs, on the other hand, are Slepian-Wolf encoded using rate-compatible punctured turbo codes in a bit-plane based fashion.

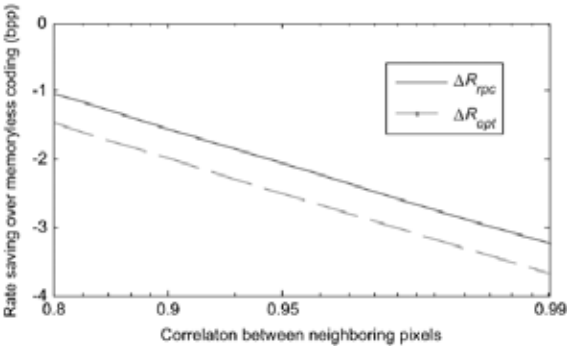

Fig. 6: Comparison of the rate saving performance of RCP and optimum coding over memoryless coding.

Table 1 Correlation Between Neighboring pixels of the Test Image

	Baboon	Lena	Peppers	Boats	Goldhill
ρ	0.81	0.97	0.98	0.97	0.99

IV. Conclusions And Future Works

In this paper, we focus on the design and analysis of a practical lossless image codec, where the image data undergoes stream-cipher based encryption before compression. We propose resolution progressive compression for this problem, which has been shown to have much better coding efficiency and less computational complexity than existing approaches. The success of RPC is due to enabling partial access to the current source at the decoder side to improve the decoder's learning of the source statistics. Our future work will focus on compression of encrypted videos, where RPC can be used for both inter-frame and intra-frame correlation learning at the decoder side.

References

[1] M.Johnson, P.Ishwar, V.M.Prabhakaran, D.Schonberg, K.Ramchandran, "On compressing encrypted data". IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[2] W. Stallings, "Cryptography and Network Security. Principles and Practice". 3rd ed. Upper Saddle River, NJ. Prentice-Hall, 2003.

[3] S. S. Pradhan, K. Ramchandran, "Distributed source coding using syndromes (DISCUS)" . IEEE Trans. Inf. Theory, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[4] J. D. Slepian, J. K. Wolf, "Noiseless coding of correlated information sources" . IEEE Trans. Inf. Theory, vol. IT-19, pp. 471–480, Jul. 1973.

[5] J. García-Frías , Y. Zhao, "Compression of correlated binary sources using turbo codes" . IEEE Commun. Lett., vol. 5, no. 10, pp. 417–419, Oct. 2001.

[6] J. Bajcsy, P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes" , in Proc. IEEE Global

[7] A. Aaron, B. Girod, "Compression with side information using turbo codes" , in Proc. IEEE Data Compression Conf., Snowbird, UT, Apr. 2002, pp. 252–261.