Concurrent Encryption and Lossless Compression using Inversion Ranks

Basar Koc bkoc@stetson.edu
Ziya Arnavut arnavut@fredonia.edu
Hüseyin Koçak hk@cs.miami.edu

For secure and efficient transmission or storage, data files are commonly compressed and encrypted. In this work, we introduce a cost-effective encryption method of files as a built-in component of a lossless compression algorithm, thus avoiding the added cost of employing two separate processes. We have shown in earlier studies that preprocessing data with Burrows-Wheeler Transformation followed by Inversion Ranking transformation in advance of the utilization of an entropy coder resulted in an extremely effective general-purpose lossless compression technique [1, 2]. During the compression process, we encrypt the frequency vector of the Inversion Ranking transformation and transmit it along with the compressed data. Since the frequency vector is required for decompression, no further encryption is necessary to secure the compressed file. Thus, encrypting only a relatively small section of data (1024 bytes) containing the frequency vector instead of the entire compressed file results in a substantial reduction in computational cost. We show in this study that the proposed concurrent encryption and lossless data compression technique is effective and resistant to common attacks using various cryptanalysis techniques on image and audio data sets.

Proposed Algorithm

The forward algorithm works as follows: First, the input file is processed with BWT. Next, IC is applied to the transformed data. The calculated inversion frequency vector F is stored for later use in encryption. To complete the compression process, the output of IC is compressed with the zero-run-length encoder (RLE-0) and the context-modeled binary arithmetic coder.

After the completion of the compression process, the remaining task is to encrypt the stored inversion frequency vector F for transmission over a non-secure channel. Depending on the intended use of the application, one can choose an encryption algorithm from various standards to encrypt the inversion frequency vector F. For experimental purposes, one such implementation of encryption of F using AES is presented in [3].

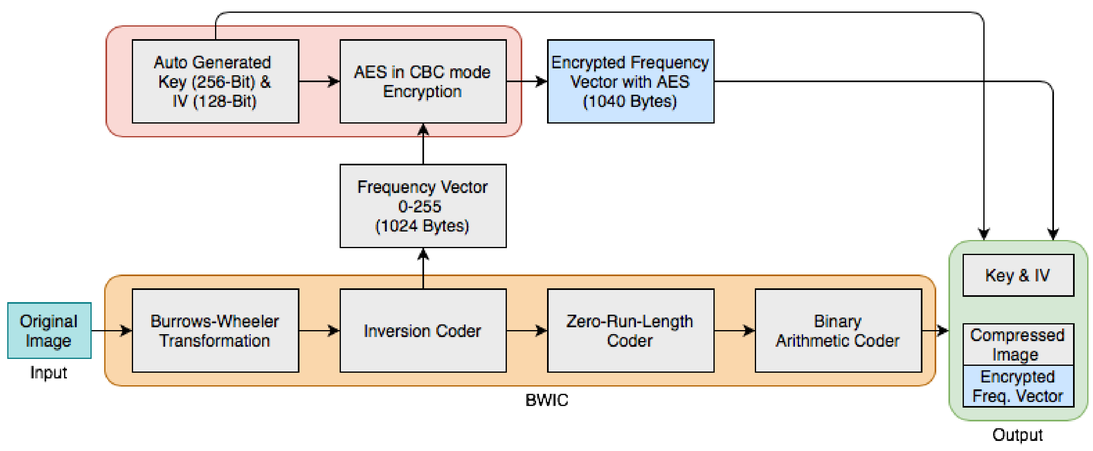


Figure 1: Schematics of concurrent encryption and lossless compression algorithm.

The salient features of our proposed concurrent encryption and compression algorithm can be recapitulated as follows:

- BWIC is an effective general-purpose lossless data compression algorithm.
- Each input data set generates its unique inversion frequency vector.
- Our proposed technique does not negatively affect compression efficiency.
- Regardless of the input data size, only the frequency vector, a small file of 1024 bytes, needs to be encrypted for transmission over a non-secure channel.

Inversion Ranking Transformation

Inversion Ranking (IR) transformation [1], also called Inversion Coding (IC), is an invertible transformation commonly used to measure sortedness of a permutation or a sequence. IR works as follows: Here, for illustration, we consider the specific input data vector

$$\bar{w} = [4, 1, 3, 1, 4, 1, 2, 4, 2, 3, 2],$$

. We scan $\bar{w}$ and create the alphabet set A which includes all the symbols (characters) used in $\bar{w}$:

$$A = \{1, 2, 3, 4\}.$$

As we scan $\bar{w}$ to construct A, we also collect the frequency of each character in A and store it in a vector F

$$F = [3, 3, 2, 3],$$

which we call the frequency vector. We hasten to point out that the number of entries of F is the same as the number of symbols in the alphabet, and the sum of the entries of F is the length of the input data vector $\bar{w}$.
Next, we compute four vectors g_i, called the inversion rank vectors, for each character A_i in the alphabet, as follows. The first entry of each vector g_i is the position index of the first occurrence of the character A_i in $\bar{w}$:

$$\bar{w} = [\underbrace{4}_{1}, \underbrace{1}_{2}, \underbrace{3}_{3}, 1, \underbrace{4}_{4}, \underbrace{1}_{5}, \underbrace{2}_{6}, \underbrace{4}_{7}, 2, 3, 2]$$

$$\begin{aligned} g_1 &= [2, \dots] \\ g_2 &= [7, \dots] \\ g_3 &= [3, \dots] \\ g_4 &= [1, \dots]. \end{aligned}$$

For each character A_i, we calculate the inversion rank (distance) between the A_i and the next A_i in $\bar{w}$, where the inversion rank is the number of elements that are greater than A_i between two consecutive A_is in $\bar{w}$. This way, we obtain all the entries of the four vectors

$$\begin{aligned} g_1 &= [2, 1, 1] \\ g_2 &= [7, 1, 1] \\ g_3 &= [3, 2] \\ g_4 &= [1, 0, 0]. \end{aligned}$$

Note that the entries of the frequency vector F are the lengths of the vectors g_i.
After calculating the inversion ranks for each character, we concatenate all the g_i vectors to obtain the inversion vector

$$\underline{w} = [\underbrace{2, 1, 1}_{g_1}, \underbrace{7, 1, 1}_{g_2}, \underbrace{3, 2}_{g_3}, \underbrace{1, 0, 0}_{g_4}].$$

As the output of the IR process, the algorithm provides the alphabet A, the frequency vector F, and the inversion vector $\underline{w}$. From F, we can determine the size of $\underline{w}$, and from $\underline{w}$ we can identify all the g_i's.

Experimental Results

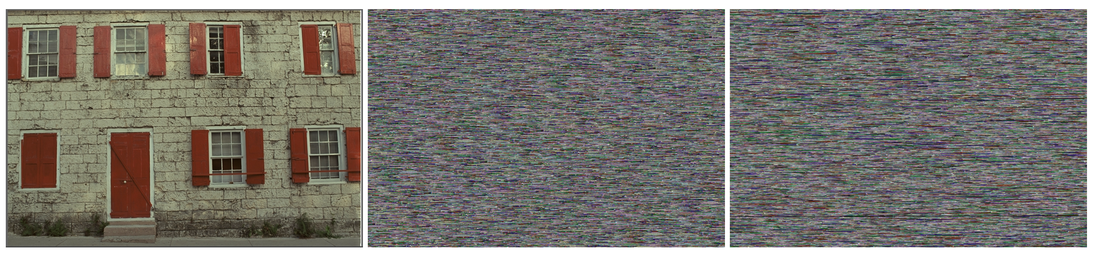


Figure 2: (a) Original Kodak01 (768×512). (b) Encrypted Kodak01 (after BWT+IC) when decrypted with *swap* perturbation of $(33, 34)$ entries of the correct key (inversion frequency vector). (c) Encrypted Kodak01 when decrypted with ± 1 perturbation of two adjacent entries of the correct key.

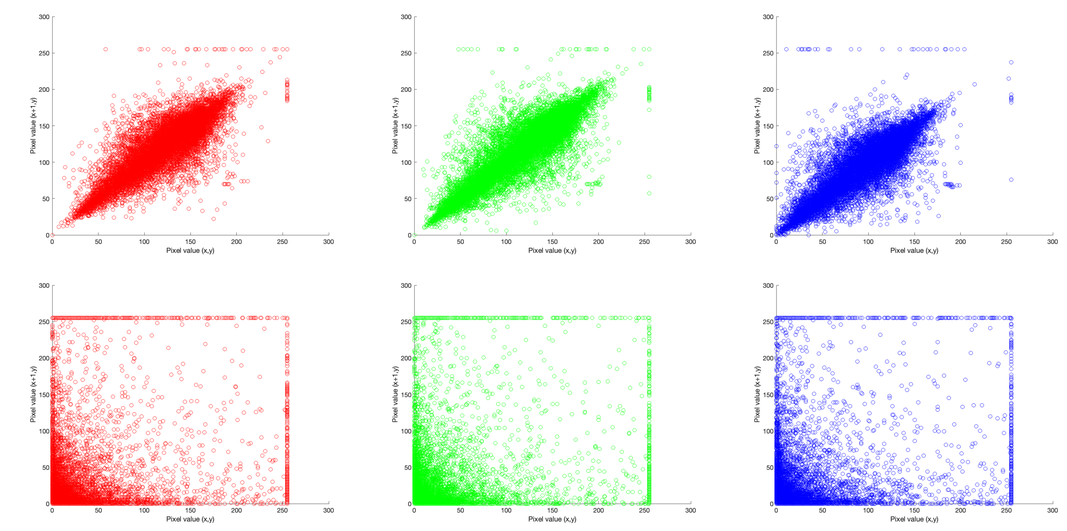
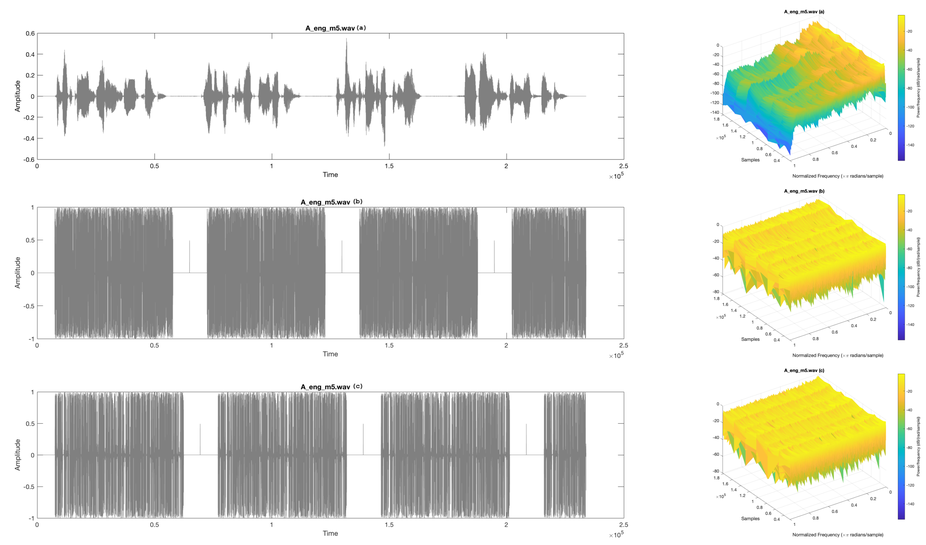


Figure 3: The six plots depict the Pearson correlation charts (using horizontal neighbor to the right) of the RGB channels of the original Kodak01 and their ciphered images. It is evident that while the pixels of the original images are highly correlated, their ciphered counterparts are decorrelated.



	Pearson Correlation	MSE
swap	-0.0023	0.1047
± 1	-2.2884e-04	0.1064

Figure 4: (a) Original audio signal (A_eng_m5.wav) and its spectrogram. (b) Decrypted with *swap*. (c) Decrypted with ± 1. The table contains the Pearson Correlation and MSE values of the original signal and the decrypted signal with the wrong keys (correct key altered by *swap* or ± 1 of indices 18 and 19)

Conclusions

Encrypting only a relatively small section of processed data (1024 bytes) containing the frequency vector instead of the entire compressed file results in a substantial reduction in computational cost. We show that the proposed concurrent encryption and lossless data compression technique is effective and resistant to various commonly employed attacks, such as brute force, histogram, correlation. In particular, we prove that the keyspace is sufficiently large and demonstrate the key sensitivity. The test data sets and the executable code of the proposed algorithm are available at [3].

References

[1] Z. Arnavut, "Inversion coding," *The Computer Journal*, vol. 47, no. 1, pp. 46–57, 2004.

[2] B. Koc, Z. Arnavut, D. Sarkar, and H. Koçak, "Technique for lossless compression of color images based on hierarchical prediction, inversion, and context adaptive coding," *Journal of Electronic Imaging*, vol. 28, no. 5, pp. 1 – 11, 2019.

[3] B. Koc, Z. Arnavut, and H. Kocak, "The executable code of concurrent encryption and lossless compression (cec)," 2021. [Online]. Available: https://www.basarkoc.com/research/cec