Color Coded Cryptography

Abstract— Data encryption technique converts data into a unreadable format so as to protect the information from external intrusion.It is thus useful in ensuring the privacy and security of the information transferred or shared between the systems. Text compression algorithms can be used to compact the text stored in the file and reduce the size of the file. It helps to reduce the consumption of resources, such as hard disk space or transmission bandwidth. Here, we propose a color coding scheme that can be used for data encryption which represents text in the form of colored blocks by grouping together binary bits and assigning them colors along with Huffman encoding scheme which is used for lossless text compression. The tandem of the above helps provide a solution to both of the problems above, as illustrate d under.

Index Terms—decryption, encryption, huffman encoding, information, security, lossless & compression

1. INTODUCTION

Color Coded Encryption is a technique of implementing a symmetrical system for security purpose. The symmetric-al system is implemented by encryption of text by converting it into image format. To reduce the size of the image file, compression algorithms are to be implemented at the encryption stage.
The converse process is used to generate at the destination system to recover the data in the original format.

Information security is the protection of information and minimises the risk of exposing information to unauthorised parties from disclosure, modification, and destruction of data. Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. The security of cipher text is totally dependent on two things: the power of the cryptographic algorithm and the confidentiality of the key. Many researchers have modified the existing algorithms to fulfil the need in the current market, yet the ciphers are vulnerable to attacks.

2. PROCEDURE FOR COLOR CODED ENCRYPTION

2.1 Data Encryption

Data encryption refers to mathematical calculations and algorithmic schemes that transform plaintext into cipher text, a form that is non-decipherable to unauthorized parties. The recipient of an encrypted message uses a key which triggers the algorithm mechanism to decrypt the data, transforming it to the original plaintext version.

The data is converted into code which usually is considered as an algorithm which uniquely represents symbols from source alphabet, by encoded strings, which may be in a target.

ASCII-based encoding using colors

The following method of ASCII-based encoding scheme using colors which was used by [1]. In RGB-256 color mode, a pixel is represented by 24 bits, in which 8 bits represent the intensity of each color. For example, a color (80, 121, 150) is represented as (01010000 01111001 10010110) . Taking the first 8 bits i.e. in this case, 01010000: ignoring the MSB bit, the remaining 7 bits or first 128 parts of the color is used to denote a character in the ASCII table. In this way, three different characters can be denoted by a single color. Thus, a text document is converted into an encoded file filled with colored dots. By using the above concepts of encoding, large amounts can be compressed and transmitted in a more secured way. This concept of ASCII- based method for representing characters and encoding them into colors is used in the proposed system.

2.2 Lossless text compression

Text compression requires that the combination of compression and decompression methods to be lossless, or else the data cannot be restored in original format. The design of data compression schemes involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression algorithm), and the computational resources required to compress and uncompressed the data.

The following method is used for lossless text compression in the Color Coded encryption at the source end:

Huffman Encoding

The Huffman encoding technique works by creating a binary tree of nodes. These can be stored in a regular array, the size of which depends on the number of symbols, n. A node can be either a leaf node or an internal node. Initially, all nodes are leaf nodes, which contain the symbol itself, the weight (frequency of appearance) of the symbol and optionally, a link to a parent node which makes it easy to read the code (in reverse) starting from a leaf node. Internal nodes contain symbol weight, links to two child nodes and the optional link to a parent node. As a common convention, bit '0' represents following the left child and bit '1' represents following the right child. A finished tree has up to n leaf nodes and n − 1 internal nodes. A Huffman tree

Huffman encoding is useful as it reduces the average code length used to represent the symbols of an alphabet.

Huffman Code Complexity

- Algorithmic complexity – O(n^2)
- Inserting a new node – O(log(n))
- n nodes insertions – O(n log(n))
- Retrieving 2 smallest node weights – O(log(n))

3. PROPOSED SYSTEM

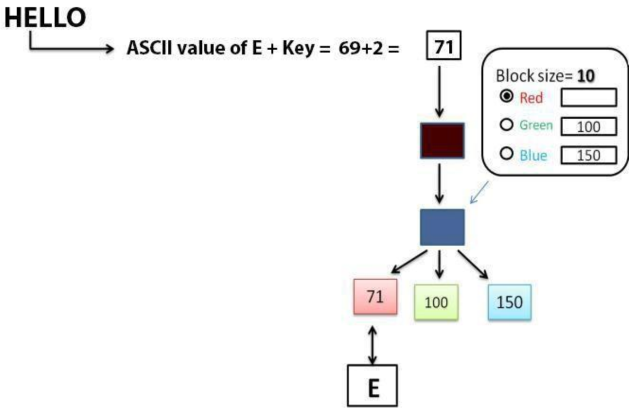

Fig 3.1: Description of working of the system

The problem definition is to design a system capable of per-forming lossless data compression on binary data using encryption and decryption. It should also function as a solution to the data protection needs of the user, holding a significant role in environments where privacy of data is critical thus contributing to information security. The encoding and compressing schemes need to be computationally and functionally efficient and must look up to provide an optimal solution to the above mentioned problems.

The system must be capable of taking input in the form of text files whose binary representation is processed and there-by encrypted in a color image. Appropriate compression methods (Huffman encoding) are used on the encrypted data to ensure a suitable trade-off between the tasks performed and space complexity issues involved. The design should achieve the best possible compression ratio, with the limited resources of a present-day PC. As a result, there are strict constraints on the memory usage and the compression speed of the design. The system presently aims to work with text file in standard ASCII-based format.

The proposed system works on two sides: *On the Senderside*, this technique compresses the file using Huffman encoding and then encodes the binary file data into a color code encrypted JPEG image file. While *On the Receiver side*, it does the reverse i.e. decrypts the image and then decompresses it, bringing back the binary text file.

3.1 SENDER SIDE

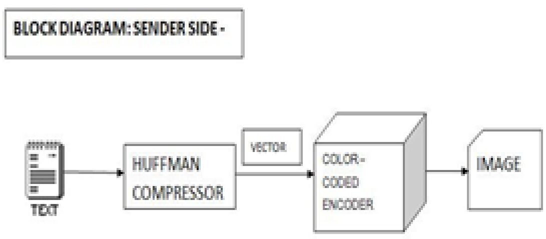

Fig 3.2: Block Diagram of Encryption system on the Sender side for Text files.

Figure 3.1 shows the block diagram of the Encryption system on the Sender side. Given a stored text file that is to be encrypted and compressed, the system first converts the file into its binary representation.

This is then given to the Huffman encoding which performs compression on the binary data by transforming into a vector.The vector is next given as input to the encryption process, which then takes in the numeric vector data and transforms it into a color coded JPEG image.

The encrypted image has a series of colored blocks. The process of color assignment is pre-decided and is done by grouping 3 bits of the binary data steam together at a time, thus giving a possibility of 8 colors in all.

3.2 RECEIVER SIDE

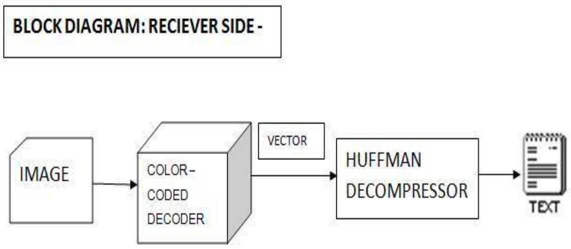

Fig 3.3: Block Diagram of Decryption module on the Receiver side for Text files.

On the receiving side, the system takes in the encrypted image as input, which it accesses for restoration. The system checks the color of the various blocks iteratively and then takes a mean of the values for getting a practical perspective of the color of the particular block in question. Using this process of decryption, it recreates the vector data. The vectored data is next given to the Huffman decoder, which converts the data back to the original binary representation as it was given in the input. The binary file is translated back to the original ASCII text file, thus restoring the text file and completing the lossless decompression on the receiver side.

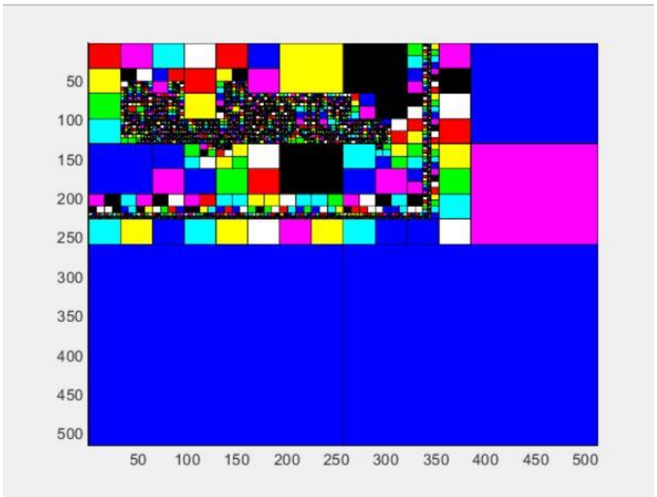

Fig 3.4: Snapshot of Color-coded block image developed on MATLAB on Sender side for Text files.

4. CONCLUSION

The system provides encryption and decryption of text files, audio and media files. The use of Huffman encoding and decoding scheme for data compression helps in dealing with the complexities of space. The proposed system works on two sides: On the Sender side, this technique is responsible for generating JPEG image from an ASCII text file; it does this by first compressing the file using Huffman encoding scheme. The vector generated by the compression is then encoded into a color code encrypted JPEG image file. On the Receiver side, it does the reverse i.e. decrypts the image which gives back the encoded vector and then decompresses it, bringing back the ASCII text file. The major advantage of the proposed system is data security. Depending on the availability of time and resources, we hope to work on the future enhancements for this system. With the assistance of a good printer and high calibrated scanner, this system can be extended to incorporate a hard copy version of the encrypted information which can be scanned at the receiver's end using a high-end scanner as shown in [9]. Another extension can be the use of the system on other formats of files such as audio and video files, images etc.

5. TECHNOLOGIES USED

Java is the powerful and object-oriented language. The primitive data types required for the functioning of process has been supplemented in Java by extensive libraries of *reference types* that are tailored for a large variety of applications. In this section, we consider reference types for string processing and image processing.