Optimal Secret Text Compression Technique for Steganographic Encoding by Dynamic Ranking Algorithm

Jagan Raj Jayapandiyan[1], Dr. C. Kavitha[2], Dr. K. Sakthivel[3]

[1]Research Scholar, Department of Computer Science, Periyar University, Salem, Tamil Nadu, India.

[2]Assistant Professor, Department of Computer Science, Thiruvalluvar Goverment Arts College, Tamil Nadu, India.

[3]Professor, Department of Computer Science and Engineering, K. S. Rangasamy College of Technology, Tamil Nadu, India.

Abstract. In this research work, proposing an algorithm, which dynamically selects the best compression algorithms among several compression techniques for steganography encoding. Ranking and selection of best algorithm for each and every steganographic transaction is based on several factors like type of cover image being used for the transmission, length of the secret message, type of the message, compression ratio of the secret message being shared, encoding ratio of secret message over the medium etc., The proposed algorithm dynamically ranks and selects the right compression algorithm to be used for the given secret file to occupy lesser embedding space in stego-image.

1. Introduction

The word ‘Steganography’ was derived from Greek words, ‘stegos’ and ‘grayfia’, which translates to ‘covered writing’, or ‘hidden writing’. Steganography is an art and science of camouflage the secret text or data, by embedding it into a media file. This media file could be an image, video or file type. Steganography and Cryptography are two different pillars for information security and enables the sender to transmit the message securely from source end to destination. Being discussed that, these methods do differ in the way of enabling the security on data. Cryptography techniques make the data to unreadable form on which the reader cannot understand the content. On the other hand, Steganography hides the existence of secret data existence itself. In basic terms, cryptography is writing an email in confidential language: people can read it but wouldn’t be able to translate what it exactly refers to. However, the existence of a (probably secret or data) message would be very obvious to anyone who sees the email. Steganography conceals the existence of secret text, while transmission and it makes any network intruder to find hard on existence of secret message.

2. Steganography and Compression

2.1. Steganography

Steganography can be performed on multiple media file. Based on the secret data that is being hidden in the stego-media or the cover image, the method and approach involved are different. By strong steganography method, a secret message can be sent over the transmission medium in more secure way and this type of concealing a secret text also avoid man in the middle type of networks attacks. As mentioned in the Figure.1, based on the steganographic data and the stego image, type of the steganography can vary, and the same could follow different operation procedure for encoding and decoding the secret message.

Major classifications of steganography are revolving around the type of stego-image being used for the steganography process [1] and the classifications can be listed as

- Text Steganography
- Image Steganography
- Audio or Video Steganography

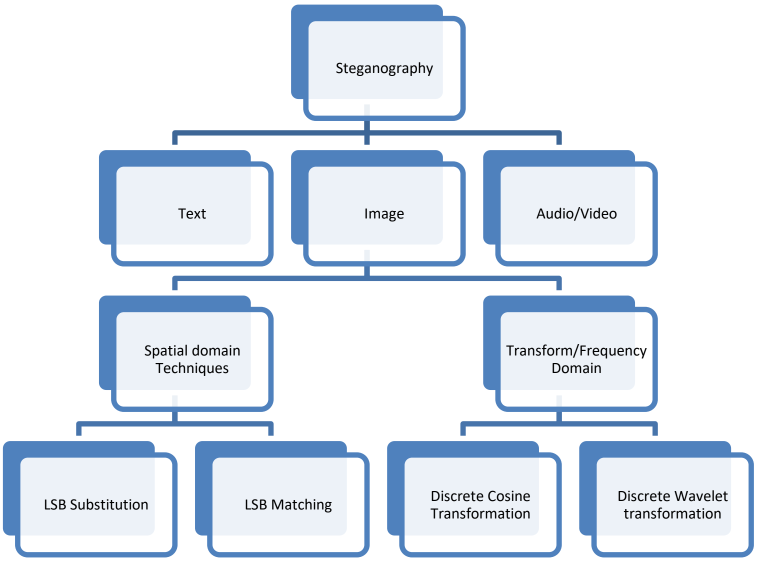
Steganography
Text
Image
Audio/Video
Spatial domain Techniques
Transform/Frequency Domain
LSB Substitution
LSB Matching
Discrete Cosine Transformation
Discrete Wavelet transformation

Figure 1. Classification of Steganography

2.2. Data Compression

Mathematical equations are used in electronic data file compression and decompression to achieve storing data in small storage space, and also a very high transmission speed. Two types of compression algorithms are

- Lossless Compression: In which a decompressed file is exactly the same size as the original (uncompressed) file because the compression-decompression process does not sacrifice any data or information.
- Lossy Compression: In which a decompressed file is smaller than the original file because some data or information (which may not be evident to the user) is lost in the process of compression-decompression.

Though compression was tried [2] and [3] along with Steganography there was no optimal way for finding right algorithm with higher compression ratio and good compression bandwidth.

2.3. Steganography Phases

- Any Steganography technique has to undergo three different phases in its life span to call the process as complete. The different phases of a typical steganography [4] and where the phases take place are as below. Sender: Encoding the secret message in stego-medium or the cover image
- Communication Channel: Transmitting the encoded stego medium
- Receiver: Decoding the secret message/text at the receiving end

Figure. 2 explains about the different phases involved on image-based steganography along with the places, where exactly it has been taking place.

$$\textit{Stego-Image } (I_S) + \textit{Secret message } (M_s) \Rightarrow \textit{Encoded Stego Image } (I_e) \quad (1)$$

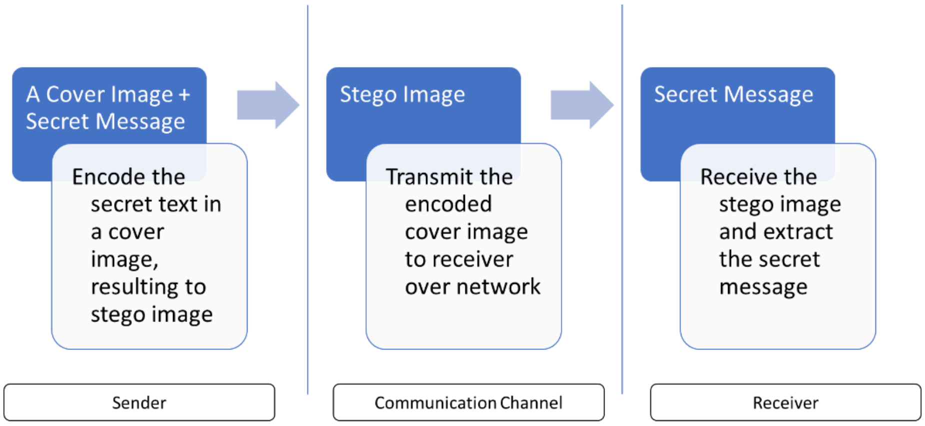
A Cover Image + Secret Message
Encode the secret text in a cover image, resulting to stego image
Stego Image
Transmit the encoded cover image to receiver over network
Secret Message
Receive the stego image and extract the secret message
Sender
Communication Channel
Receiver

Figure 2. Phases of image-based steganography process

3. Compression types and determining right compression

3.1. BZip2 (.bz2)

BZ2 extension refers to a pure data compression format that does not provide archival features, the BZip2 compression algorithm is based on Burrows–Wheeler transform.

Compression rate is rather slower than in zip format and gzip format, which employs classic 'Deflate' algorithm (even if properly implemented Bzip2 algorithm can be conveniently rendered parallel, and benefiting from recent multi-core CPU), but faster than more efficient compression methods as in RAR format, 7Z format, and new ZIPX format. Also, the compression ratio is typically between older ZIP / GZ files based on Deflate and modern 7Z and RAR formats.

Because of the design limitation, BZip2 compression on Unix based systems has been usually cascaded into TAR archiving multiple data and metadata files (file attributes, date/time etc) are merged into a single uncompressed.tar container generating TBZ2 files that can be recognized as TBZ, TB2, or *.TAR.BZ2/*.tar.bz2 file extensions.

- Usage: BZip2 compression is typically used to archive data and metadata on Unix and Unix-like systems and can also be used as an alternative compression algorithm in ZIP and 7Z files –resulting archives can be read from most file archives. If a better compression than classic deflate-based ZIP / GZIP compression is required, it can be suggested, and in all situations an algorithm on memory and CPU faster and less heavy than LZMA-based 7Z or PPMd-based RAR compression is preferable.

3.2. 7-Zip (.7z)

7Z extension refers to the popular 7-Zip archiving format, a new file archiving program written by Igor Pavlov and published as freeware and open source software. 7-Zip (sometimes incorrectly pronounced in 7zip) was originally designed by the p7zip (POSIX-7Zip) project team on Unix based systems for Microsoft Windows systems.

Spanned multi-volume 7z archives are named by.001 file extension, incrementing file length numerator for each subsequent volume of the collection-spanned 7z files are break byte-wise, consistent with Unix based "split" command.

- Usage: 7z is a good choice whenever it is necessary to achieve a higher compression ratio, and solid encyclical support for keeping archived information private. When there is sufficient time for compression or backup, this technique is suggested over zip format to help reduce the output volume. Few tools like peazip [10] also allows the development of 7z-format self-extracting archives (SFX), so when downloading one of those self-extracting files, the user do not need a 7-Zip compliant extractor.

RAR file format is renowned for its compression ratio and superior to zip, comparable to 7z format and zipx format-being the Lempel-Ziv (LZSS) algorithm-based compression technique and partial matching (PPMd, Dmitry Shkarin) projection.

3.3. ZIP (.zip)

ZIP extensions are common Microsoft Windows device standard for archiving files. Different file spanning methods are used by different tools to build broken multi-volume zip archives, the most common being raw file spanning (as applied to the 7-Zip or Unix / Linux "split" command) and personalized spanning of WinZip. Modern archive formats such as 7Z (7-Zip/p7zip), RAR (WinRar), and ACE (WinAce) have gained in popularity and implemented many enhancements (some of which were actually introduced in ZIP format) such as improved compression ratio, recovery records, heavy file encryption, etc. often at lower speed and higher memory demands compared to slower, lightweight Deflate compression used in .zip format.

- Usage: Zip is a good option whenever it is essential to keep the archive consistent with most recipients of archive managers, allowing it an excellent choice for the delivery of content, as it is generally possible to unzip it on any platform. For the same reason, long-term archive / backup processing is also suggested, as such a ubiquitous format (with requirements for the public domain) is unlike any conceivable situation. Furthermore, for both archiving and extraction steps, ZIP is usually significantly faster than more efficient compression formats.

3.4. RAR (.rar)

RAR specification also provides solid file encryption by design RAR4 archives rely on AES-128 derived encryption. Encrypted RAR5 archives based on AES-256 in CBC mode, with improved key scheduling Notably,.rar format offers advanced error correction and data recovery functionality allowing optional recovery records during archive development (for example from WinRAR). RAR4 and RAR5 have the same .rar extensions as the format revision rate is specified in the header field. Spanned multi-volume rar archives are built by extension R01-incrementing the file extension number for the following set volumes.

4. Compression algorithm selection attributes

Below are the criteria for selecting a compression algorithm for data compression.

4.1. Compression Bandwidth

How much of data can be taken for compression for a given time (usually represented in MB/s or KB/s like units). Higher the value is considered as best for usage.

$$CB_c = size(M_s) / CT_m \tag{2}$$

Where,

CB_c = Compression Bandwidth

M_s= Secret Message (or) Uncompressed message

CT_m = Compression time

4.2. Compression Ratio

If it does not make our data smaller than the uncompressed size, there is really no point. Compression Ratio tell how well the data is squeezed to occupy less space [5]. Higher the value is considered as best for usage.

$$CR_m = size(M_s) / size(M_c) \tag{3}$$

Where,

CR_m = Compression Ratio

M_s= Secret Message (or) Uncompressed message

M_c = Compressed Message

4.3. Compression Time

This factor tells how quick the data can be compressed and represents in any unit of time. Higher the value is considered as best for usage.

$$CT_m = size(M_s) / CB_m \tag{4}$$

Where,

CT_m = Compression Time

M_s= Secret Message (or) Uncompressed message

B_c = Compression Bandwidth

4.4. Space Savings

Defined as the size reduction of size in compressed file compared to the uncompressed value[6]. Higher the percentage is considered as best for usage.

$$SS_m \text{ (in \%)} = 1 - (size(M_c)/size(M_s)) \tag{5}$$

Where,

SS_m = Space savings

M_c = Compressed Message

M_s= Secret Message (or) Uncompressed message

5. Proposed ranking algorithm for optimal compression algorithm determination

Below proposal addresses the best algorithm selection based on the criteria discussed in Section IV. This proposal addresses the user's problem of selecting right algorithm when the stego image is constrained by space [12]-[16] or whether the network is limited for data transfer. The algorithm selects the top three algorithms with higher compression ratio and then the one which have best compression bandwidth.

5.1. Dynamic Ranking Algorithm

```
function data_compression_select($M_s$,$A_s$);
input: One secret text file $M_s$ and Compression algorithms list $A_s$, where As is a list
output: Two sorted list based on $CR_m$ and $CB_m$
set i = 0
loop (for all the elements in $A_s$)
        $M_{c[i]}$ = compress($M_s$)
        result_dict ($A_{s[i]}$,key(CR) = comp_ratio($M_s$,$A_{s[i]}$)
        result_dict ($A_{s[i]}$,key(CB) = comp_bw($M_s$,$A_{s[i]}$)
        result_dict ($A_{s[i]}$,key(CT) = comp_time($M_s$,$A_{s[i]}$)
        result_dict ($A_{s[i]}$,key(SS) = comp_time($M_s$,$A_{s[i]}$)
      set i = i + 1
set sort_list($CR_m$) = sort(result_dict ($A_{s[i]}$,keys($CR_m$))
set sort_list($CB_m$) = sort(result_dict ($R_{c[0-2]}$,keys($CB_m$))
return (sort_list($CB_m$)[n-1])
```

Algorithm 1. Dynamic Ranking Algorithm

6. Experimental Results of Proposed Algorithm

6.1. Sample data

In this sample calculation, we are taking below data for the compression algorithm ranking and selection for secret text compression

- Compression Algorithms Set (As): Four major data compression algorithms for the study such as ARC, RAR, Bzip2 and 7z
- File Sizes (Ms): This algorithm is exercised on various file sizes, ranging from 7kb to 6mb
- Message Files (MS): Message content has been taken into considering four sample text files, which are large_6mb.txt, medium_sized_3mb.txt, text_sample_10kb.txt and smaller_7kb.txt

6.2. 7z

On implementing above proposed algorithm (Algorithm 1) using 7z data compression method yielded the below results for ranking. The results for (2), (3), (4) and (5) are:

Table 1. Attributes Values for 7z Data Compression Algorithm

File Name	large_ 6mb.txt	medium_sized_3 mb.txt	small_2mb.t xt	smaller_7kb.t xt
Uncompressed Message Size in bytes (M_s)	6,488,666	3,071,342	2,097,152	6,223
Compressed Message Size in bytes (M_c)	1,808,683	851,788	2,097,415	1,566
Compression Time in ms –(CT_m)	2600	1300	476	559
Comp. Bandwidth in KB/s (CB_c)	2437	2307	4302	10
Compression Ratio - CR_m	3.59	3.61	1.00	3.97
Secret message size saved in % (SS_m)	72%	72%	0%	75%

6.3. ARC

On implementing above proposed algorithm (Algorithm 1) using ARC data compression method yielded the below results for ranking

Table 2. Attributes Values for ARC Data Compression Algorithm

File Name	large_ 6mb.txt	medium_sized_ 3mb.txt	small_2mb.t xt	smaller_7kb.t xt
Uncompressed Message Size in bytes (M_s)	6,488,666	3,071,342	2,097,152	6,223
Compressed Message Size in bytes (M_c)	1,548,414	736,839	2,090,005	2,366
Compression Time in ms –(CT_m)	3345.00	784.00	473.00	442.00
Comp. Bandwidth in KB/s (CB_c)	1894	3825	4329	13
Compression Ratio - CR_m	4.19	4.17	1.00	2.63
Secret message size saved in % (SS_m)	76%	76%	0%	62%

6.4. Bzip2

On implementing above proposed algorithm (Algorithm 1) using BZIP2 data compression method yielded the below results for ranking

Table 3. Attributes Values for Bzip2 Data Compression Algorithm

File Name	large_ 6mb.txt	medium_sized _3mb.txt	small_2mb.t xt	smaller_7kb.t xt
Uncompressed Message Size in bytes (M_s)	6,488,666	3,071,342	2,097,152	6,223
Compressed Message Size in bytes (M_c)	1,765,784	808,540	2,106,503	1,511
Compression Time in ms –(CT_m)	1300	611	850	345

Comp. Bandwidth in KB/s (CB_c)	4874	4908	2409	17
Compression Ratio - CR_m	3.67	3.80	1.00	4.12
Secret message size saved in % (SS_m)	73%	74%	0%	76%

6.5. RAR

On implementing above proposed algorithm. [8] (Algorithm 1) using RAR data compression method yielded the below results for ranking

Table 4. Attributes Values for RAR Data Compression Algorithm

File Name	large_ 6mb.txt	medium_sized_3 mb.txt	small_2mb.t xt	smaller_7kb.t xt
Uncompressed Message Size in bytes (M_s)	6,488,666	3,071,342	2,097,152	6,223
Compressed Message Size in bytes (M_c)	2,168,302	808,540	2,106,503	1,511
Compression Time in ms –(CT_m)	5230	670	503	123
Comp. Bandwidth in KB/s (CB_c)	1211	4476	4071	49
Compression Ratio - CR_m	2.99	3.80	1.00	4.12
Secret message size saved in % (SS_m)	67%	74%	0%	76%

7. Comparison charts for compression attributes

Below chart (Figure. 3) depicts the compression bandwidth of all the above sample files on various compression techniques selected for this experimental analysis.

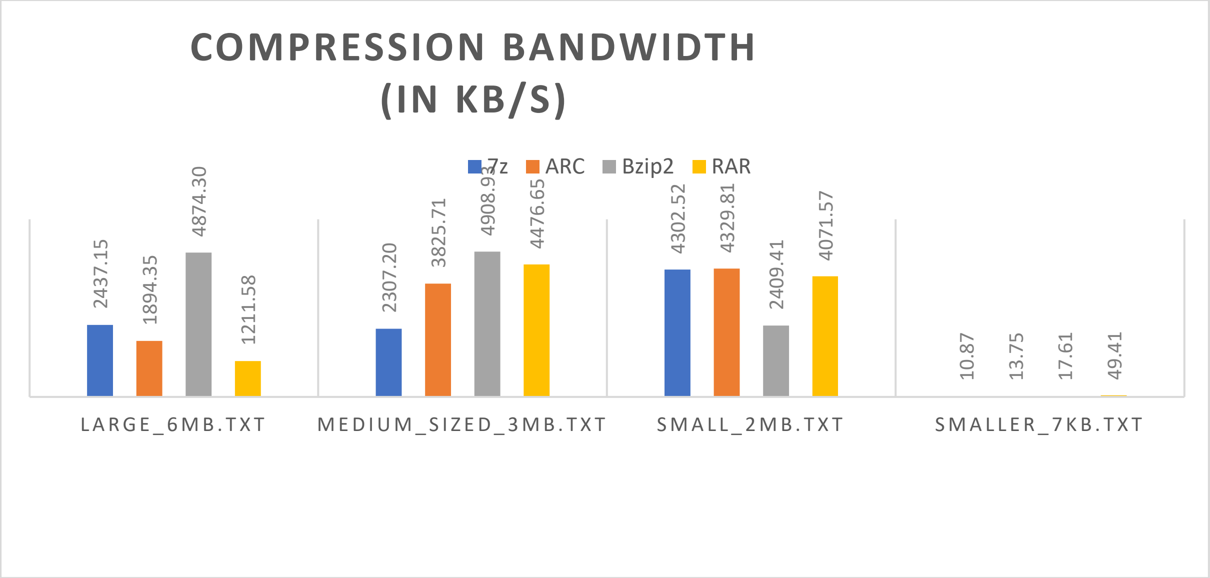

Figure 3. Compression Bandwidth (CB_m) comparison chart

The chart in Figure.4 compares the compression ratio (CR_m) attribute [9] for all the compression algorithms in the sample data taken

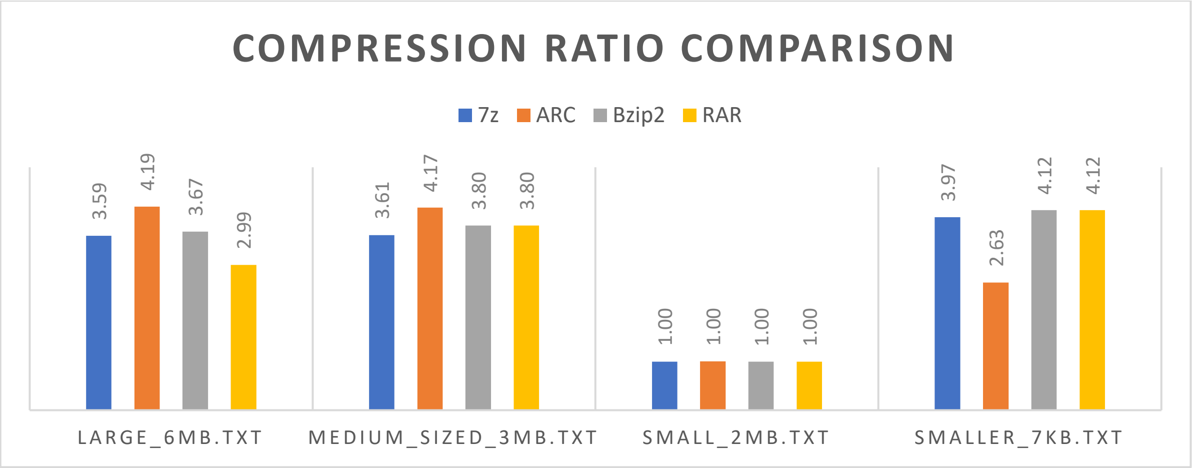

Figure 4. Compression Ratio (CR_m) comparison chart

Similarly, the following chart Figure. 5 compares the space savings (SS_m) attribute for all the sample files on various compression techniques selected in this experimental analysis.

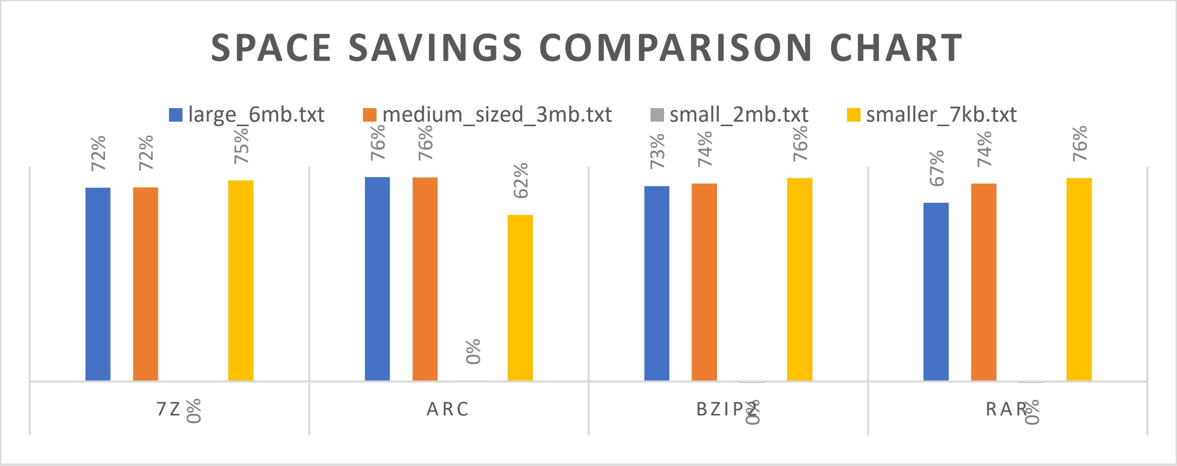

Figure 5. Compression Bandwidth (SS_m) comparison chart

Below table (Table 5.) explains the ranking of compression attributes and associated ranks, which are calculated through above proposed algorithm. In this example, selecting ARC method for data compression will need the lesser compute power for stego encoding and lesser bandwidth in communication channel.

Table 5. Compression Ratio Value Comparison for large_6mb.txt message file

Algorithm	Compression Ratio (CR_m)	Index	Rank based on proposed algorithm
7z	3.59	0	3
ARC	4.19	1	1
Bzip2	3.67	2	2
RAR	2.99	3	4

Table 6. Compression Bandwidth Value Comparison for large_6mb.txt message file

Algorithm	Compression Bandwidth in KB/s (CB_c)	Index	Rank based on proposed algorithm
7z	2437	0	2

ARC	1894	1	3
Bzip2	4874	2	1
RAR	1211	3	4

Selecting Bzip2 method based on the below table (Table 6.) for the compression while stego coding will need the lesser compute resource, while compressing the secret message.

8. Conclusion

Selection of right data compression algorithm and associating them with the steganography method is always a key for optimal memory and better stego cover image utilization. Proposed research work will enable the sender and receiver to utilize the lesser compute power for stego encoding/decoding and lesser compute resource. Based on the experimental results and observations gathered in the experiment, this mechanism of ranking and dynamic selection of data compression algorithm will ensure the reduced space utilization in the stego object, through which more payload can be embedded in the secret image and compute resource is also consciously used.

9. References

[1] Thangadurai, K., Sudha Devi, G. “An analysis of LSB based image steganography techniques,” International Conference on Computer Communication and Informatics (ICCCI), IEEE, Oct. 2014

[2] R. Mishra, A. Mishra and P. Bhanodiya, "An edge based image steganography with compression and encryption," 2015 International Conference on Computer, Communication and Control (IC4), Indore, 2015, pp. 1-4.

[3] Carpentieri, B, Castiglione, A, De Santis, A, Palmieri, F, Pizzolante, R. Compression-based steganography. Concurrency Computat Pract Exper. 2019

[4] Jagan Raj, S Prasath, “Validating Data Integrity in Steganographed Images using Embedded Checksum Technique. International Journal of Computer Applications, 2015

[5] A. Yazdanpanah and M. R. Hashemi, "A new compression ratio prediction algorithm for hardware implementations of LZW data compression," 2010 15th CSI International Symposium on Computer Architecture and Digital Systems, Tehran, 2010, pp. 155-156.

[6] P. A. Alsberg, "Space and time savings through large data base compression and dynamic restructuring," in Proceedings of the IEEE, vol. 63, no. 8, pp. 1114-1122, Aug. 1975.

[7] 7-Zip, 07 2017, [online] Available: http://www.7-zip.org/

[8] Y. Wei, N. Zheng and M. Xu, "An Automatic Carving Method for RAR File Based on Content and Structure," 2010 Second International Conference on Information Technology and Computer Science, Kiev, 2010, pp. 68-72

[9] T. Suzuki and K. Hayashi, "Text data compression ratio as a text attribute for a language-independent text art extraction method," 2010 Fifth International Conference on Digital Information Management (ICDIM), Thunder Bay, ON, 2010, pp. 513-518

[10] PeaZip, [online] Available: http://www.peazip.org/

[11] Data Compresison, Compression Ratio, Space saved during data compression [online] https://en.wikipedia.org/wiki/Data_compression

[12] S. Khan, M. A. Irfan, M. Ismail, T. Khan and N. Ahmad, "Dual lossless compression based image steganography for low data rate channels," 2017 International Conference on Communication Technologies (ComTech), Rawalpindi, 2017, pp. 60-64.

[13] I. G. Wiryawan, Sariyasa and I. G. A. Gunadi, "Steganography based on least significant bit method was designed for digital image with lossless compression technique," 2018 International Conference on Signals and Systems (ICSigSys), Bali, 2018, pp. 98-102.

[14] A. Darbani, M. M. AlyanNezhadi and M. Forghani, "A New Steganography Method for Embedding Message in JPEG Images," 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 2019, pp. 617-621.

[15] K. Rajalakshmi and K. Mahesh, "Video steganography based on embedding the video using PCF technique," 2017 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, 2017, pp. 1-4.

[16] S. L. Chikouche and N. Chikouche, "An improved approach for lsb-based image steganography using AES algorithm," 2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B), Boumerdes, 2017, pp. 1-6.

[17] D. Kaur, H. K. Verma and R. K. Singh, "A hybrid approach of image steganography," 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 1069-1073.

[18] Y. Yiğit and M. Karabatak, "A Stenography Application for Hiding Student Information into an Image," 2019 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, Portugal, 2019, pp. 1-4.