JES Journal of Engineering Sciences

Vol 10, Issue 11, Nov /2019
ISSN NO: 0377-9254

MULTI-LEVEL ADAPTIVE COMPRESSION TECHNIQUE FOR PAYLOAD ENCODING IN STEGANOGRAPHY ALGORITHMS

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

[1]*Research Scholar, Dept. of Comp. Science, Periyar University, Salem, T.N, India*

[2]*Asst Prof, Dept. of Comp. Science, Thiruvalluvar Govt. Arts College, T.N, India,*

[3]*Professor, Dept of CSE, K. S. Rangasamy College of Technology, TN, India*

***Abstract*— This research work recommends a method that adaptively chooses the strongest compression algorithms for steganography encoding amongst several compression methods. Selection of the best method for every secret file type is always based on several factors such as the type of cover image being used for communication, size of the secret message being transferred, message file type, compression ratio of the shared secret message file, the compression ratio of the secret message to the stego medium, etc. This proposal provides a holistic solution to handle compression techniques for all the secret message file types.**

***Keywords*—Steganography, Compression, ranking, dynamic selection, Information Security**

I. INTRODUCTION

The term Steganography derives from the Greek words "stegos" and "grayfia," meaning "covered writing" or "writing secretly" Steganography is an art and science to camouflage a secret text or data by embedding it in a media file. The content being embedded could be of any type like text, image, video or any other media file. By using strong steganography, a secret content can be sent more safely through a transmission medium, which also prevents people from attacking via network. Steganography and cryptography are two different information security principles that allow the recipient to securely communicate the message from the origin to the destination. Being discussed that, these methods do differ in the way of enabling the security on data. These methods differ in the way data protection is enabled. The tools of cryptography make the data unreadable, and the content cannot be comprehended by the reader. Steganography, on the other hand, conceals the presence of secret data. Essentially, cryptography writes a secret email, people can read it but cannot understand what it relates to precisely. Nevertheless, it would be very clear to everyone who sees the e-mail that the message exists (certainly secret or data). Steganography masks the inclusion of hidden messages and makes it hard to find the secret text for any network interferer or network analysis method.

II. STEGANOGRAPHY AND COMPRESSION

A. Steganography

Steganography can be done in various media formats. The method and approach are different depending on the secret data that are concealed in the stego-media or cover image. As mentioned in Fig.1, the Steganography type may differ and the same process for encoding and decoding the hidden message may follow based upon the steganographic data and stego image.

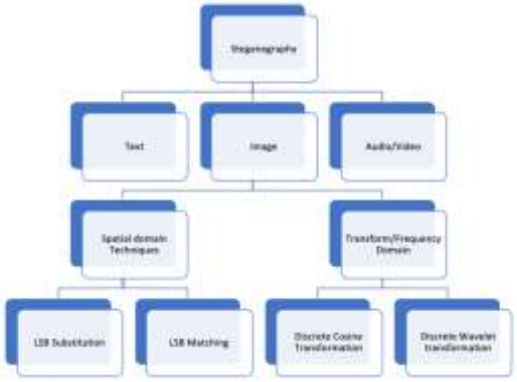


Fig. 1. Classifications of Steganography

The key classifications of steganography apply to the kind of stego-image used for steganography [1] and can be categorized as:

- Text Steganography
- Image Steganography
- Audio or Video Steganography

B. Data Compression

Mathematical formulas are used to compress and decompress electronic data files to reach storages and transfer speeds in little storage space. Two types of algorithms being commonly used compression, which are:

- *Lossless Compression*: In which the file decompressed is the same as the (uncompressed) file because no data or

information is lost during the compression-decompression process.

- *Lossy Compression:* If the file is smaller than the original, a decompressed file because some data or information has been lost (which the user may be unaware of).

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.

C. Steganography Phases

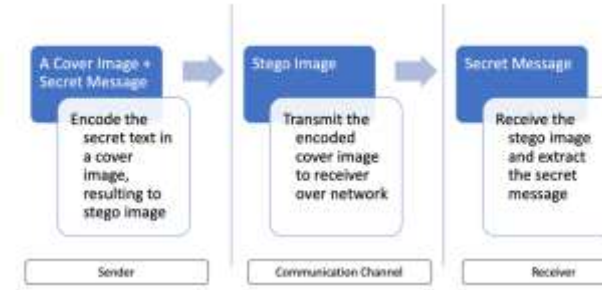
Any Steganography technique needs to undergo three different life stages in order to call a full cycle as given below:

- 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/file at the receiving end

Fig. 2 explains about the various phases involved in image steganography together with the places, where it took place exactly.

Stego-Image (I_S) + Secret message (M_s) => Encoded Stego Image (I_e)

Fig. 2. Phases of image-based steganography process



III. COMPRESSION TYPES AND DETERMINIING RIGHT COMPRESSION

A. BZip2 (.bz2)

The BZ2 extension is based on Burrows-Wheeler transformation and refers to a pure data compression format that does not include any archival functionality.

Compression speed is rather slugger than in zip and gzip formats that use classical' deflate' algorithm (though a Bzip2 algorithm can be easily run in parallel with the current multi-core CPU); but it is slower than compression methods that are more effective than those of RAR, 7Z and ZIPX formats in new formats. The compression ratio between old and new deflate-based ZIP / GZ files is usually 7Z / RAR formats.

The compression of BZip2 on Unix systems has been cascaded into TAR archiving due to the design limitations - multiple data and meta files are merged (file attributes, date / time etc.,) into a single uncomprimed.tar container - generating TBZ2 files that can be recognized as TBz, TB2, or * .TAR.BZ2 or *.tar.bz2 extensions.

- *Usage:* BZip2 compression is usually used for archiving data and metadata on Unix and Unix-like systems and also for the alternative compression algorithm in Zip and 7Z files. If better compression is required than traditional deflate-based ZIP / GZIP compression, a memory and CPU algorithm faster and heavier than 7Z or PPMd-based RAR compression can be suggested in all circumstances.

B. 7-Zip (.7z)

The 7Z extension refers to Igor Pavlov's popular 7-Zip archiving format, which was released as freeware and open source software. The p7zip (POSIX-7Zip) group of the Unix based systems for Microsoft Windows systems was initially designed with 7-Zip (sometimes mis-prompted as 7zip).

The .001 file extension is the starting name of a spanned multi-volume 7z archive; the incremental file length numerator is a byte-based breakdown, consistent with a Unix based split command.

- *Usage:* 7z is a good decision wherever a higher compression ratio is required, and a strong encyclical support for privacy. This technique is preferred over zip format when there is sufficient time for compression or backup to help reduce the output size. Few tools like peazip[2] are also available to develop 7z self-extracting (SFX) files. Therefore, the user does not have to have a 7-Zip compatible extractor when downloading a single auto-extractor file.

C. ZIP (.zip)

Microsoft Windows generic system for archiving files are common extensions to ZIP. Different processes are used to build broken multi-volume zip archives, the most frequent of these are raw file scope (as used with the command 7-Zip and Unix / Linux "split"). Modern formats of archive such as 7Z (7-Zip / p7zip), RAR (WinRar) and ACE (WinAce) have gained popularity and have implemented a lot of enhancements (some of which were actually ZIP formats), such as an enhanced compression, retrieval registries, heavy file encryption.

- *Usage:* Zip is good when it is important to keep the archive in line with most of the archive managers ' recipients, so the archive provides an excellent choice in terms of delivering content, because it can usually be removed on any platform. The same is valid, as such an omnipresent format (with public domain requirements) is unique to any conceivable situation for long-term archives /

backup storage. Moreover, ZIP is typically much faster than more effective compression formats for both archiving and extraction.

D. ARC (.arc)

ARC extension refers to the native archive format designed for FreeArcwhat, a modern open source file archiver built by Bulat Ziganshin with a high Compression Bandwidth(11 lossless data compression algorithms and filters are assisted, including LZMA and PPMD) with good performance equivalent to or stronger than rar format..

IV. COMPRESSION ALGORITHM SELECTION ATTRIBUTES

The parameters for choosing a good data compression algorithm are as below:

A. Compression Bandwidth

Compression bandwidth is defined as the value that represents how much data can be used for compression for a certain period of time (usually in units like MB / s or KB / s). Higher the value is considered as best for usage.

$$CB_c = size(M_s) / CT_m$$

Where,

CB_c = Compression Bandwidth

M_s= Secret Message (or) Uncompressed message

CT_m = Compression time

B. Compression Ratio

Compression Ratio expresses on how well the data is squeezed to occupy less space [5]. Higher the value is considered as best for usage.

$$CR_m = size(M_s) / size(M_c)$$

Where,

CR_m = Compression Ratio

M_s= Secret Message (or) Uncompressed message

M_c = Compressed Message

C. Compression Time

This parameter tells how quick the data can be compressed and the unite represents in time. Lesser the value is considered as best for usage.

$$CT_m = size(M_s) / CB_m$$

Where,

CT_m = Compression Time

M_s= Secret Message (or) Uncompressed message

B_c = Compression Bandwidth

D. 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.

$$SS_m (in\ \%) = 1 - (size(M_c)/size(M_s))$$

Where,

SS_m = Space savings

M_c = Compressed Message

M_s= Secret Message (or) Uncompressed message

V. PROPOSED MULTI-LEVEL RANKING ALGORITHM FOR 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 or whether the network is limited for data transfer. This proposed algorithm selects the top 'n' algorithms with higher compression ratio and then the one which have best compression bandwidth among the n elements.

A. Multi level Ranking Algorithm

Algorithm 1. Multi level Ranking Algorithm

```
function data_compression_select(Ms,As);
Input: One secret file Ms and Compression
algorithms list As, where As is a list
output: one algorithm based on CR and CB
values
set i = 0
loop (for all the elements in As)
        Mc[i] = compress(Ms)
        result_dict(As[i],key(CR) =
comp_ratio(Ms,As[i])
        result_dict(As[i],key(CB) =
comp_bw(Ms,As[i])
        result_dict(As[i],key(CT) =
comp_time(Ms,As[i])
        result_dict(As[i],key(SS) =
comp_time(Ms,As[i])
        set i = i + 1
set sort_list(Rc) = sort(result_dict (As[i],keys(Rc))
set sort_list(Bc) = sort(result_dict (Rc[0-
2],keys(Bc))
return (sort_list(Bc)[n-1])
```

VI. EXPERIMENTAL RESULTS OF PROPOSED ALGORITHM

A. Sample data

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

- *Compression Algorithms Set (A_s):* Four major data compression algorithms for the study such as ARC, RAR, Bzip2 and 7z
- *File Sizes for sample dataset:* This algorithm is exercised on various file sizes, ranging from 6 kb to 10 mb
- *Message Files (M_s):* Message content has been taken into considering four sample secret files of different file types, which are
    - iso_8859-1.txt [12],
    - marbles.bmp [13],
    - file_example_WAV_10MG.wav [14],
    - file_example_MP4_640_3MG.mp4 [15]

B. 7z

On implementing above proposed algorithm (Algorithm 1) using 7z data compression method yielded the below results for ranking

TABLE I. ATTRIBUTES VALUES FOR 7Z DATA COMPRESSION ALGORITHM

File Name	iso_8859-1.txt	marbles.bmp	file_example_WAV_10MG.wav	file_example_MP4_640_3MG.mp4
Uncompressed File Size in KB (M_s)	6	4,165	10,163	3,042
Compressed Message Size in KB (M_c)	2	1,787	6,886	2,625
Time taken in Seconds (CT_m)	0.532	5.500	4.700	1.700
Compression Bandwidth(KB/s) (CB_c)	1.9	320	1400	1403
Compression Ratio(CR_m)	3.00	2.33	1.48	1.16
Secret message size saved (SS_m)	67%	57%	32%	14%

C. ARC

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

TABLE II. ATTRIBUTES VALUES FOR ARC DATA COMPRESSION ALGORITHM

File Name	iso_8859-1.txt	marbles.bmp	file_example_WAV_10MG.wav	file_example_MP4_640_3MG.mp4
Uncompressed File Size in KB (M_s)	6	4,165	10,163	3,042
Compressed Message Size in KB (M_c)	2	1,816	5,910	2,642
Time taken in Seconds (CT_m)	1.600	1.300	0.874	0.354
Compression Bandwidth(KB/s) (CB_c)	3	1204	6634	7286
Compression Ratio(CR_m)	3.00	2.29	1.72	1.15
Secret message size saved (SS_m)	67%	56%	42%	13%

D. Bzip2

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

TABLE III. ATTRIBUTES VALUES FOR BZIP2 DATA COMPRESSION ALGORITHM

File Name	iso_8859-1.txt	marbles.bmp	file_example_WAV_10MG.wav	file_example_MP4_640_3MG.mp4
Uncompressed File Size in KB (M_s)	6	4,165	10,163	3,042
Compressed Message Size in KB (M_c)	2	1,762	9,250	2,618
Time taken in Seconds (CT_m)	0.250	0.969	3.602	1.200
Compression Bandwidth(KB/s) (CB_c)	6	1708	2458	1946
Compression Ratio(CR_m)	3.00	2.36	1.10	1.16
Secret message size saved (SS_m)	67%	58%	9%	14%

E. ZIP

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

TABLE IV. ATTRIBUTES VALUES FOR RAR DATA COMPRESSION ALGORITHM

File Name	iso_8859-1.txt	marbles.bmp	file_example_WAV_10MG.wav	file_example_MP4_640_3MG.mp4
Uncompressed File Size in KB (M_s)	6	4,165	10,163	3,042
Compressed Message Size in KB (M_c)	2	2,303	9,460	2,620

Time taken in Seconds (CT_m)	0.244	0.762	0.761	0.495
Compression Bandwidth(KB/s) (CB_c)	6	2970	12288	5120
Compression Ratio(CR_m)	3.00	1.81	1.07	1.16
Secret message size saved (SS_m)	67%	45%	7%	14%

TABLE V. COMPRESSION RATIO VALUE COMARISION FOR FOR MARBLES.BMP MESSAGE FILE

Algorithm	Compression Ratio (CR_m)	Index	Rank
7z	2.33	0	2
ARC	2.29	1	3
Bzip2	2.36	2	1
ZIP	1.81	3	4

Selecting Bzip2 method based on the above table (Table-V) yields better compression ratio. After the compression process, secret message file (M_s) takes lesser space while embedding the same in stego image or cover image.

TABLE VI. COMPRESSION BANDWIDTH VALUE COMARISION FOR FOR MARBLES.BMP MESSAGE FILE

Algorithm	Compression Bandwidth in KB/s (CB_c)	Index	Rank
7z	3200	0	1
ARC	1204	1	3
Bzip2	1708	2	2

On the top three ranked algorithms from Table-V considered for next level of ranking to conclude the algorithm to be used for embedding if compute resource is a constraint during compression. Ranking the output of Table-V based on compression bandwidth will yield Table-VI, ranking the same will give 7z as the best compression algorithm. It can be concluded that the proposed algorithm selects the right algorithm for stego secret file compression technique based on multi-level ranking.

VII. CONCLUSION

Selecting and associating the right data compression algorithm with the steganography approach is always a key to optimum memory and better use of stego-cover image. The recommended research work would allow the sender and receiver to use the lesser computational power for steganographic encoding / decoding and less computational energy. Based on the results of the experiments and findings gathered in the experiment, this multi-level ranking and dynamic data compression algorithm selection process would ensure a reduced use of storage in the stego object, by means of which more payload can be inserted in the hidden image and less use of the software resource.

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] Text file, iso_8859-1.txt, https://www.w3.org/TR/PNG/iso_8859-1.txt

[13] Image file, marbles.bmp, https://www.fileformat.info/format/bmp/sample/index.htm

[14] Audio file, file_example_MP3_2MG.mp3, https://file-examples.com/wp-content/uploads/2017/11/file_example_MP3_2MG.mp3

[15] Video file, file_example_MP4_640_3MG.mp4, https://file-examples.com/wp-content/uploads/2017/04/file_example_MP4_640_3MG.mp4