A Novel Steganography Technique That Embeds Security Along With Compression

Anuradha *anu107sharma@gmail.com*
Student/CSE DCRUST,Murthal
Sonepat, 131039, India

Nidhi *nidhi.sharma.1012@gmail.com*
Student/IT Banasthali Vidyapeeth
Bansathali, 304022, India

Rimple *rimple.gilhotra@hotmail.com*
Student/CSE Banasthali Vidyapeeth
Bansathali, 304022, India

Abstract

Problem faced by today's communicators is not only security but also the speed of communication. This paper provides a mechanism that increases the speed of communication by reducing the size of content; for this data compression method is used and security factor is added by using Steganography. Firstly, the focus has been made on Data Compression and Steganography. Finally, proposed technique has been discussed. In Proposed technique first data is compressed to reduce the size of the data and increase the data transfer rate. Thereafter on compressed data state table operation is applied to improve the security. Then, this is used as the input to the LSB technique of Steganography. At receiver end, the LSB extraction technique is used, thereafter the state table operation in reverse form is applied and finally the original data is obtained. Hence our proposed technique is effective that can reduce data size, increases data transfer rate and provides the security during communication.

Keywords: Compression, Arithmetic Coding, Steganography, Hexadecimal, One Time Pad, Least Significant bit (LSB).

1. INTRODUCTION

The present network scenario demands exchange of information with more security and reduction in both the space requirement for data storage and the time for data transmission [9]. This can be accomplished by compression and data hiding. Now a day's people use the network as the basic transport for transmitting their personal, secure and day to day information. Hence they need some form of security from the third person during transmission which is provided by Steganography where Steganography refers to the science of invisible communication [10, 11]. Along with that sometimes the data that is to be sent is in huge amount that requires lots of space and bandwidth, so we must have a mechanism with the help of which we can reduce the size of data as well as time and bandwidth is saved; this is done by using Arithmetic Coding.

Proposed mechanism embeds the compressed data behind the cover data; this mechanism is used to achieve the present network scenario for exchange of information with more security and compression.

2. DATA COMPRESSION

Compression is used just about everywhere. Compression algorithms reduce the redundancy in data representation to decrease the storage required for that data. The task of compression consists of two components, an encoding algorithm that takes a message and generates a "compressed" representation and a decoding algorithm that reconstructs the original message or

some approximation of it from the compressed representation. We distinguish between lossless algorithms, which can reconstruct the original message exactly from the compressed message, and lossy algorithms, which can only reconstruct an approximation of the original message. Lossless algorithms are typically used for text, and lossy for images and sound where a little bit of loss in resolution is often undetectable, or at least acceptable [4].

2.1 Arithmetic Coding

In information theory an entropy encoding is a lossless data compression scheme that is independent of the specific characteristics of the medium. One of the main types of entropy coding assigns codes to symbols so as to match code lengths with the probabilities of the symbols. Typically, these entropy encoders are used to compress data by replacing symbols represented by equal-length codes with symbols represented by codes where the length of each codeword is proportional to the negative logarithm of the probability. Therefore, the most common symbols use the shortest codes. The most popular entropy encoding is Arithmetic Encoding [5].

In arithmetic coding, a message is represented by an interval of real numbers between 0 and 1. As the message becomes longer, the interval needed to represent it becomes smaller, and the number of bits needed to specify that interval grows. Successive symbols of the message reduce the size of the interval in accordance with the symbol probabilities generated by the model. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the message.[6] Before anything is transmitted, the range for the message is the entire interval [0, I), denoting the half-open interval 0 <= x < 1 [12]. Thus, the algorithm successively deals with smaller intervals, and the code string, viewed as a magnitude, lies in each of the nested intervals. The data string is recovered by using magnitude comparisons on the code string to recreate how the encoder must have successively partitioned and retained each nested subinterval. [13]

3. STEGANOGRAPHY

Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected [1]. Steganography involved a Greek fellow named Histiaeus. As a prisoner of a rival king, he needed a way to get a secret message to his own army. He shaves the head of a willing slave and tattoos his message on to the bald head. When hairs grew back, off he went to deliver the hidden writing in person [3].

Steganography derives from the Greek word “steganos” means covered or secret and “graphy” means writing. On the simplest level Steganography is hidden writing, whether it consists of invisible ink on paper or copyright information hidden within an audio file. Today, Steganography, “stego” for short, is most often associated with in other data in an electronic file. This is done by replacement the least important or most redundant bits of data in the original file i.e. bits that the human eye or ear hardly misses with hidden data bits [2].

3.1 Where it Comes From [3,8]

One of the earliest uses of Steganography was documented in histories. Herodotus tells how around 400 B.C. Hisitieaus shaved the head of his most trusted slave and tattooed it with the message which disappeared after the hair has regrown. The purpose of this message was to investigate the revolt against the Persians. Another slave could used to send the reply.
During the American Revolution, invisible ink which would glow over a flame was used by both the British and the American’s to communicate secretly. German hides text by using invisible ink to print small dots above or below letters and by changing the heights of letter-strokes in cover texts.

In world war 1prisoners of war would hide Morse code messages in letters home by using the dots and dashing on I, j, t and f. censors intercepting the messages were often altered by the phrasing and could change them in order to alter the message.

During world war 2nd the Germans would hide data in microdots. This involved photographing the message to be hidden and reducing the size so that it can be used as a period within another document.

3.2 Types of Steganography[3]

Steganography can be split into two types, these are Fragile and Robust. The following section describes the definition of these two different types of Steganography.

3.2.1 Fragile

Fragile Steganography involves embedding information into a file which is destroyed if the file is modified. This method is unsuitable for recording the copyright holder of the file since it can be so easily removed, but is useful in situations where it is important to prove that the file has not been tampered with, such as using a file as evidence in a court of law, since any tampering would have removed the watermark. Fragile Steganography techniques tend to be easier to implement than robust methods.

3.2.2 Robust

Robust marking aims to embed information into a file which cannot easily be destroyed. Although no mark is truly indestructible, a system can be considered robust if the amount of changes required to remove the mark would render the file useless. Therefore the mark should be hidden in a part of the file where its removal would be easily perceived.

3.3 Key Features

There are several key features regarding Stegangraphy and its usage are as follows:

- The main goal of Steganography is to hide a message m in some audio or video (cover) data d, to obtain new data d', practically indistinguishable from d, in such a way that an eavesdropper cannot detect the presence of m in d'.
- The goal of Steganography is to hide the message in one-to-one communication
- We can hide as much data as possible.
- Ease of detection level should be Difficult.
- We can hide as much data as possible.
- Goal of detector is to detect the hidden data.

3.4 Applications

There are various areas where Steganography can be applied:

- Confidential communication and secret data storing.
- Protection of data alteration
- Access control system for digital content distribution.
- Media Database systems
- Corporate espionage, Cover Communication by Executives, Drug dealers, Terrorists.

4. PROPOSED TECHNIQUE

The proposed technique is based on the concept of arithmetic coding and Steganography in which a word of text is converted into floating point number that lie in range between 0 and 1. This floating point number is converted into hexadecimal number and after that one time pad and a state table is used to hide the compressed hexadecimal data.
At Receiver end, data is extracted by using the Steganography method that will be explained later; after that decompression is done to obtain the original word.

4.1 Compression and Hiding

Firstly input symbol is compressed using arithmetic coding after that one time pad and the state table is used on the result of arithmetic coding.

ALGORITHM

To compress and encrypt the message Algorithm includes following steps:

Step 1: Using table encode the input symbol.

a) Initialize lower_ bound=0, upper_ bound=1

b) While there are still symbols to encode

Current _range = upper _bound - lower _bound
Upper_ bound = lower _bound + (current _range * upper _bound of new symbol)
Lower_ bound = lower_ bound + (current _range * upper_ bound of new symbol)

End while

Step 2: The string may be encoded by any value within the probability range and after that convert the output decimal number into hexadecimal data.

Step 3: Choose 2nd MSB of the selected cover image. This is the one time pad.

Step 4: Now, the state table operation is applied on the hexadecimal equivalent and the one time pad. The information about this state table is exchanged between sender and receiver earlier. This state table will help in confusing the intruder because the intruder does not know anything about the state table. Hence, the security level is increased further. The state table is given in Table 1. [7]

Input		**Output**	
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1

TABLE 1: State Table

Step 5: The output obtained from step 4 is used in LSB substitution method of Steganography.

Step 6: The final embedded cover image is send to the receiver side.

4.2 Decompression and Extraction

Algorithm

Step 1: Extract the LSB's from the cover image.

Step 2: Choose 2nd MSB's of the cover Image, this is the onetime pad.

Step 3: Apply the state table (Table 1) to the LSB's and the 2nd MSB's of the cover image.

Step 4: The output obtained from step 3 is the original hidden data in hexadecimal format.

Step 5: Convert the hexadecimal format into decimal equivalent.

Step 6: Apply arithmetic decoding procedure.

Encoded_ value=Encoded input

While string is not fully decoded

Identify the symbol containing encoded value within its range
current_ range = upper _bound of new symbol - lower _bound of new symbol
encoded value = (encoded _value - lower_ bound of new symbol) ÷ current_ range

End while

Output: The output is the original symbol.

4.3 Example
Suppose Input Data is: “ganga”

Step 1: Create Probability Table

For character g:

Occurrence of character ‘g’ in Input data is “2”.
Probability is 2/5=0.4

For character a:
Occurrence of character ‘a’ in Input data is “2”.
Probability is 2/5=0.4

For character n:
Occurrence of character ‘n’ in Input data is “1”.
Probability is 1/5=0.2
The probability table is prepared according to the occurrences of the letters. This is explained in table 2.

Symbol	Probability	Range(lower_ bound, upper_ bound)
A	40%	[0.00,0.40)
G	40%	[0.40,0.8)
N	20%	[0.8,0.1)

TABLE2: Symbols along with probability of occurrence

4.3.1 Compression and Hiding
Data to be encoded is “ganga”

Step1:
Encode 'g'
current_ range = 1 - 0 = 1
upper bound = 0 + (1 × 0.4) = 0.4
lower bound = 0 + (1 × 0.8) = 0.8

Encode 'a'
current range = 0.8 - 0.4 = 0.4
upper bound = 0.4 + (0.4 × 0.0) = 0.4
lower bound = 0.4 + (0.4 × 0.8) = 0.56

Encode 'n'
current range = 0.56-0.4 = 0.16
upper bound = 0.4 + (0.16 × 0.8) = 0.528
lower bound = 0.4 + (0.16 × 1) = 0.56

Encode 'g'
current_ range = 0.56-0.528 = 0.032
upper bound = 0.528 + (0.032 × 0.4) = 0.5408
lower bound = 0.528 + (0.032 × 0.8) = 0.5536

Encode 'a'
current range = 0.5536 - 0.5408 = 0.0128
upper bound = 0.5408 + (0.0128 × 0.0) = 0.5408
lower bound = 0.5408 + (0.0128 × 0.4) = 0.54592

Step2:

The string "ganga" may be encoded by any value within the range [0.5408, 0.54592).
Now output is 0.54260 and its hexadecimal equivalent= 01010100001001100000

Step3: Select an Image which is considered as a cover image.

11001010	10101010	11100010	10100001	11100011
11100010	10100001	10101101	10001001	10101010
10101101	10101010	10100001	11100011	10100001
11100011	11001010	10101010	11001010	11100010
10101010	10101101	10100001	10101010	11100010
10101010	11100011	11001010	11100010	10101010
10101010	11001010	10101010	10101010	11100011
10101010	11100010	11100011	10101010	11001010

TABLE 3: Cover image

Step4: Choose 2nd MSB's of cover Image as a one time pad key.

Step5: Our One time pad is – 10101100000001011011
Data- 01010100001001100000 from step 2.
Apply operation on bits according to the given state table [1].

Step6: Final Output is: 011001100111000000001000001110010100010 1

Step7: Now Apply LSB substitution method of stenography to hide data in cover image.

11001010	10101011	11100011	10100000	11100010
11100011	10100001	10101100	10001000	10101011
10101101	10101011	10100000	11100010	10100000
11100010	11001010	10101010	11001010	11100010
10101011	10101100	10100000	10101010	11100010
10101010	11100011	11001011	11100011	10101010
10101010	11001011	10101010	10101011	11100010
10101010	11100010	11100011	10101010	11001011

TABLE 4: Cover image with data hidden inside.

4.3.2 Decompression and Extraction
Step 1: Extract the LSB's of cover image which gives us hidden data.
Hidden Data: 011001100111000000001000001110010100010 1

Step2: Reverse the operation on bits by taking combination of 2 bits, which gives the combination of one time pad key and actually compressed data i.e.
10011001101100000000100001101101000 1010

Step3: Separate the one time pad key and compress data i.e.

One time pad key: 10101100000001011011
Data- 01010100001001100000

Step4: Convert hexadecimal format into decimal format i.e. 0.54260

Step5: Using the probability ranges from table decodes the three character string encoded as 0.54260.

Decode first symbol
0.54260 is within [0.4, 0.8)
0.54260 encodes 'g'

Remove effects of 'g' from encode value
Current _range = 0.8 - 0.4 = 0.4
Encoded _value = (0.54260-0.4) ÷ 0.4 = 0.3565

Decode second symbol
0.3565 is within [0.0, 0.4)
0.3565 encodes 'a'

Remove effects of 'a' from encode value
current range = 0.0 - 0.4 = 0.4
encoded value = (0.3565 - 0.0) ÷ 0.4 = 0.89125

Decode third symbol
0.89125 is within [0.8, 1)
0.89125 encodes 'n'

Remove effects of 'n' from encode value
Current _range = 1 - 0.8 = 0.2
Encoded _value = (0.89125-0.8) ÷ 0.2 = 0.45625

Decode second symbol
0.45625 is within [0.4, 0.8)
0.45625 encodes 'g'

Remove effects of 'g' from encode value
Current range = 0.0 - 0.4 = 0.4
Encoded value = (0.45625 - 0.4) ÷ 0.4 = 0.14063

Decode third symbol
0.14063 is within [0.8, 1)
0.14063 encodes 'a'
Now we are with our secret data i.e. "ganga"

5. BENEFITS

- In proposed system generated cipher text takes very less bandwidth of secure channel.
- Highly Secure.

6. CONCLUSION AND FUTURE SCOPE

The Present network scenario demands exchange of information with reduction in both space requirement for data storage and time for data transmission along with security. Our proposed technique fulfils all such requirements as this technique uses the concept of data compression and Steganography. Along with that the state table that increases the security further because the intruder does not have any idea about this state table. By using this technique we can reduce the

size of data and after that compressed data can be hidden to provide the security. Hence this technique increased the data transfer rate and security during data communication. There exists some enhancement in the compression method used as future work. We can use any other compression method that will provide better compression ratio than the existing one.

7. REFERENCES

[1]. Christian Cachin, "An Information-Theoretic Model for Steganography", A preliminary version of this work was presented at the 2nd Workshop on Information Hiding, Portland, USA, 1998, and appears in the proceedings (D. Aucsmith, ed., Lecture Notes in Computer Science, vol. 1525, Springer).Original work done at MIT Laboratory for Computer Science, supported by the Swiss National Science Foundation (SNF).March 3, 2004, pp. 1-14.

[2]. Eric Cole, Ronald L. Krutz, James W. Conley, "Network security bible" Wiley Pub. 2005, pp. 482-520

[3]. SecondLieutentJ.caldwell,"Steganography",UnitedStatesAirForce,http://www.stsc.hill.af.mil /crosstalk/2003/caldwell.pdf, June2003.

[4]. Guy E. Blelloch. Computer Science Department. Carnegie Mellon University blellochcs. cmu.edu.http://www.cs.cmu.edu/afs/cs/project/pscicoguyb/realworld/www/compression. pdf ,September 25, 2010.

[5]. V.Kavitha , K.S Easwarakumar. "Enhancing Privacy in Arithmetic Coding" ICGST-AIML Journal, Volume 8, Issue I, pp. 23-28, June 2008.

[6]. IAN H. WIllEN, RADFORD M. NEAL, and JOHN G. CLEARY. "Arithmetic coding for data compression." Communications of the ACM , Volume 30 Number 6,pp.521-540, June 1987.

[7]. Ajit Singh, Nidhi Sharma. "Development of mechanism for enhancing the Data Security using Quantum Cryptography." Advanced Computing: An International Journal (ACIJ), Vol.2, No.3, pp.22-25, May 2011.

[8]. Herodotus. The Histories. London, England: J. M. Dent & Sons Ltd, 1992.

[9]. Ajit Singh , Rimple Gilhotra. "Data security using private key encryption system based on arithmetic coding." International Journal of Network Security & Its Applications (IJNSA), vol.3, no.3, May 2011.

[10]. Mehdi Kharrazi, Husrev T. Sencar Nasir Memon. "Performance study of common image Steganography and steganalysis techniques." *Journal of Electronic Imaging 15(4),041104 (Oct–Dec 2006)*

[11]. M. Kharrazi, H. T. Sencar, and N. Memon, Image Steganography Concepts and Practice, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, Singapore _2004

[12]. J.A Storer, (1988) "Data Compression: Methods and Theory" Computer Science Press.

[13]. Glen G. Langdon, (1984) "An introduction to arithmetic coding", IBM Journal of Research and Development Volume 28, No.2