International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 08 | Aug - 2021 ISSN: 2582-3930

A REVIEW OVER TEXT BASED STEGANOGRAPHY

DR. KALPESH RASIKLAL RAKHOLIA
HOD,COMPUTER SCIENCE
PKM COLLEGE - JUNAGADH

ABSTRACT

Steganography is the craftsmanship and study of sending covered up messages. In current correspondences frameworks, this implies concealing data in correspondence media like sound, text, and pictures. Preferably, aside from the sender and collector, no outsider ought to try and associate the presence with such messages. Advanced interchanges frameworks require the utilization of blunder revising codes (ECC) to battle commotion, or mistakes, presented by the comparing (correspondence) channel. Essentially, an ECC adds repetition to a message so the mistakes presented by the channel can be remedied.

In our specific circumstance, the code repetition can be used to embed stega bits (that is, pieces of a mysterious message) covered as fake blunders, which thusly, can't be recognized from certified channel mistakes. Subsequently, loud correspondence channels give a reasonable system to steganography. In this work, we centre on text-based steganography. The fundamental ECC is the Golay code, what separates the data succession into squares of 12 pieces. Toward the finish of the encoding interaction, every 12-cycle block is changed into a 23-piece block, called a code word. The Golay code is equipped for amending up to three blunders in a square of 23 pieces and is alluring for battling mistakes in extremely uproarious correspondence channels. Two methods of inclusion of stega pieces are talked about and looked at.

The modes address a compromise among exactness and mystery. In the initial, a more exact form of the mysterious message is recuperated in correlation with the second; nonetheless, it is more defenceless to being recognized by a busybody than the subsequent mode.

INTRODUCTION

Steganography is an art or science of transmitting hidden messages. In modern communication system, this means hiding information in communication media such as audio, text and images. Steganography is supposed to be originated from Greek culture where the Greek word steganos means concealed and graphein means to write.

Techniques for hiding information have existed for centuries. In Ancient Greece, secret messages were written on wooden plates and wax was used to cover them. Methods include writing hidden messages on paper written in invisible ink in the blank spaces of the papers. This technique was adopted quite successfully during World War II by the French.

Some other techniques were also implemented in the past. Messages were written on the back of postage stamps. Germans used microdots during World War I and World War II. Microdots are nothing but a text or an image

substantially reduced in size onto a disc of 1 mm in diameter. Special cameras were used to generate microdots attached to letters. These microdots usually went unnoticed for any intruders and could easily read by the authorized recipient microscope. Techniques such as spread spectrum are used these days in digital communication. Electromagnetic or acoustic signals generated for a specific bandwidth are spread over a much wider bandwidth to avoid signal interference or signal jamming.

Digital watermarking is also one of the many applications of steganography. Visible watermarks are used for copyright protections and source tracking but in case of invisible watermarking, the information is difficult to perceive. The secret message is hidden in a digital signal. The spread spectrum mentioned earlier is used for audio watermarking. Spread Spectrum is used to embed watermarks which can be implemented easily in any time domain. After spreading the spectrum the information is hidden in the form of a watermark and is added to the sender signal as a watermarked signal. The core principal for steganography is that apart from the sender and the receiver, no third party or the intruder can suspect the presence of any such hidden or covert message.

This phenomenon clearly distinguishes steganography from a very renowned technique of information hiding which is cryptography. In cryptography, the information is hidden by doing encryption of the original message by various encryption algorithms. This encryption process converts the plain text into a cipher text with the help of the encryption key. If ever a third party intrudes and manages to extract the cipher text, this encrypted message is hard to decode without the key. This clearly states that in Cryptography the third party can detect the presence of secret message easily though it may or may not decrypt the encoded message which is disparate from the principal of steganography which hides the message as well as the presence of the message. Therefore where cryptography protects the content of the message, steganography protects both message and communicating parties. So except the authorized persons, no third party can ever think of any such secret message in the communication. Hence steganographic communications do not attract attention since they are never highlighted or encrypted but always hidden.

In computer systems as well, steganography is extensively used. Pictures are embedded in video material. Secure shell connections, remote desktop software such as telnet, virtual host always include some amount of delay before sending the information packets over the network. These delays can be used to encode data. Texts are hidden in web pages. Information is concealed within computer files which can be audio files, jpeg images or bit mapped images which are larger in size and contain lot of information in it. For example, every nth color bit is replaced with some message bit and sent over the transport network. This change is so minute that it usually goes unnoticed due to highly redundant code stream. Some tools can be used to transmit valuable data in normal network traffic. Internet Control Message Protocol (ICMP) is an Internet protocol used for networked computers to send error messages for diagnostic or routing purposes in IP datagram. These ICMP messages are part of the IP header and transmitted the resulting datagram. Linux has a ping utility which adds 56 bytes of ICMP message to the existing header. Loki is another such tool that hides data in ICMP traffic. Loki is a client-server program which can be used to transmit data

secretly across the network through back door into a Unix system. A directory starting with dot (.) is a hidden directory. A directory starting with three dots (…) can be created to store secret files so they do not come into the file lists. On Windows systems, the C,/winxp/system32 or C,/winnt/system directories are where all the Windows .dll, .dib and

set up files are placed. This directory can be used to securely store all the covert files assuming that no one really dares to tamper or touch the files in those important directories.

REQUIREMENTS

The primary aim behind developing this tool is to understand how steganography can be achieved using error correcting codes for very noisy communication channels with cover media being a text file. Upon completion, this tool will help the students from mathematics and statistics as well as computer science in learning more about error correcting codes and their applications and different techniques used in Information security branch for secure and covert data communication. This software can also be integrated into different text editors such as office word, Kwrite etc. for creating documents with secret messages embedded. The requirements gathered have been further classified into platform requirements and functional requirements.

PLATFORM REQUIREMENTS

• The main objective in choosing the software development kit (software language) was that it should be platform independent so that final product will have the capability to run on any environment irrespective of the operating system.

• The software should run as a stand-alone application rather than a web based software. Hence Java SDK instead of Java Enterprise Edition has been chosen to be the appropriate language for writing the code. The operating system is Ubuntu Linux considering in mind the importance of open source software.

• There should be a facility to store different versions of code and some repository where code can be checked in and checked out. This is applicable whenever any modifications are made or any new feature gets added. Keeping this in mind, SVN (subversion) repository has been used.

FUNCTIONAL REQUIREMENTS

• The software should be able to read any large text file as a cover media so that secret message can be embedded into it.

• The Error correcting code (ECC) being chosen should not induce additional complexity to the existing bit stream.

• The ECC should not increase the bandwidth of the channel by adding too many redundant bits in such a way that the performance and efficiency gets hampered.

• The core principle of steganography should be achieved. That is, message as well as its existence should be concealed.

• The transportation of the message blocks over the communication channel should not take large amounts of time.

• The decoder version of the software should be capable enough to join all the received and decoded data blocks to get back the original secret message is received.

• Since the Golay code is used as an ECC which can correct up to three errors per 23 bit codeword, there should be a facility for the encoder to select how many artificial errors (stega bits) he wants to send for each codeword.

• In the Golay code mode2, the detection scheme should be intelligent enough to select the erred bits which were not picked by the normal decoding mechanism.

• There should be a facility to introduce the genuine channel errors along with the artificial errors so that the communication channel used will look more original.

NOISY CHANNELS AND ERROR CORRECTING CODES

Any message to be transmitted over the communication channel needs some level of protection. This is because of many things such as noise, channel error in the communication. These hindrances not only change the message content but also the meaning of the message very commonly known as noise.

Noise is an unwanted part of any digital or analog signal. It is the factor which is responsible for degrading the quality of the signal by acting as an interference or blockage in the communication channel. This entity is naturally present in most of the communication channels corrupting the signals passed over. Hence signal to noise ratio should be as high as possible to ensure the error free communication. However the occurrence of noise is totally random and if proper filter is not used to detect its presence, it usually goes unnoticed creating disturbances at the receiver's end. To combat the noise, various techniques are used such as increasing the power of the signal, implementing some sort of a modulation such as frequency modulation (FM), amplitude modulation (AM) etc. or adding a lot of redundant bits to the original signal. Some of these operations are expensive such as increasing the power of a signal or amplitude modulation. Adding a lot of redundant bits can also be an inefficient method since it increases the channel bandwidth beyond capacity. But if handled in a proper manner, through error correcting codes this method can be used very effectively for to detect the noisy bits or errors at the receiver's end.

A very noisy channel is an attractive medium for steganographic communications. This is achieved by having a low transmission rate of concealed messages sent block by block. This type of communication produces an almost untraceable secret message transfer.

Since these noisy channels require error correcting codes to come over the noise, the code redundancy is utilized very ingeniously to insert the secret message bits to be passed in form of artificial channel errors. The insertion of steganographic bits over a honest communication channel is possible only because of the bit redundancy created by the error correcting codes.

Error correcting codes (ECC) provide a technique for data transmission in which a few extra bits are added in each block of data in order to detect the errors and then correct those errors that may occur in the communication. These redundant bits also known as parity bits make sure that the message is received error free at the receiver's end and once the errors are corrected, these parity bits are easy to remove from the original content. Parity Bits take care of the number of checked bits in the code i.e. they are implemented as odd parity when number of 1's in a given set of bits are even and similarly for even parity.

ECC have been successfully used in physical and data link layer of the OSI model and also implemented in data disks and computer physical memories in the form of checksum. Checksum is an addition of all the codewords in a given set of bits. ECC are mainly divided into two classes viz. convolution and block codes. Convolution codes operate on bit by bit basis and block codes are processed per block basis. This thesis is focused on block codes. ECC are also known as forward error correction (FEC) since no re-transmission of the message is performed. There are many types of block codes such as repetition codes,

BCH codes, Golay codes, Hamming codes. Some of these are very efficient codes such as Golay, BCH etc. Golay Codes are encoding and decoding techniques are implemented in this thesis work. The channel used for steganographic communication is a binary symmetric channel explained below.

BINARY SYMMETRIC CHANNEL

The communication channel used in most of the steganographic communications and encoding-decoding mechanisms of Golay codes in this thesis is a binary symmetric channel (BSC). Let us assume that the communication messages are sent over a very noisy channel and that we can send only two symbols viz. 0 and 1. Also assume that when the sender sends the symbol 0, the receiver receives the same symbol 0 with probability p and receives 1 with probability q. Similarly when the sender sends symbol 1, the probability that 1 is received is p and 0 is received is q. Then for a binary symmetric channel, $p + q = 1$ also meaning

$q = p - 1$.

This type of communication is possible only in binary symmetric channel (BSC). BSC is very frequently used in information and coding theory. It is assumed that in BSC, the symbol is received with a high probability, but if the symbol or the bit gets flipped, then that probability is very small. Figure 4.1 illustrates this type of communication

channel very clearly. In this diagram, if X is a variable that is randomly transmitted and Y is the variable randomly received, then the channel is characterized by the conditional probabilities shown below:

Pr(Y = 0 / X = 0) = p (4.1)

Pr(Y = 1 / X = 0) = 1 – p (4.2)

Pr(Y = 1 / X = 1) = p (4.3)

Pr(Y = 0 / X = 1) = 1 – p (4.4)

A linear code of length n and dimension k is a subspace C of the vector space (F2)n (all n tuples with entries in F2 where F2 ={0, 1} the binary field). Such a code is referred to an (n, k) code. Elements of the code are called codewords. A generator matrix for an (n, k) code is a k x n matrix whose rows form a basis for the vector space C. This generator matrix often denoted as G, is of the form (Ik | A) where Ik is an identity matrix (k x k) and A is the standard matrix with dimensions ((n-k) x n).

The Hamming distance of a linear code C is equal to the minimum distance between any two codewords in C. This is also equal to the minimum weight of the nonzero codewords in C. These linear codes bear a special property that any two non zero codewords c Є C differ 14 in at least d positions where d is the Hamming distance between the two codewords and addition of any these codewords, e.g. c0 and c1, fetch a result say c2 which also is a codeword belonging to the same set C. In mathematical terms, Hamming distance is the number of coordinates In which c0 and c1 disagree.

Encoding of codes has been explained in brief in the previous chapter. The programming details and implementation techniques that were used in this thesis are described below. Golay codes can detect up to six errors and correct up to three errors in a codeword of 23 bits. The example considered here has a cover media as a text file and secret message as a plain text. Both the cover media and the secret message are converted into binary streams of 1 and 0. This conversion takes place using a function named readFile().

This function has been written in Java and explained below. The readfile() command takes its argument as the filename whose contents are supposed to be converted and stored into a binary stream. This stream is created by a Java class which opens the file, reads it and makes use of Java input-output functions by importing Java.io api. The large stream is first read into a variable which contains non-binary characters. This string is separated into individual characters and each character is stored into a character array as an array variable using tocharArray() function. Hence if the string is

Desktop, the character array, e.g. arr[], is created and Desktop is stored into arr[] as arr[0] = 'D', arr[1] = 'e', arr[2] = 's' etc using the tocharArray() function.

Each character has an associated integer value with it. This integer is also known as an ASCII value. For example, 'D' has an ASCII value 68, 'A' has ASCII value 65, 'B' is 66, 'e' is 101, number '0' is 48 and Space is 32. The character array elements are converted into the respective integer values using typecasting technique of Java. Typecasting changes the data type of a variable to another data type. Here a character is converted to integer by simply specifying the character as (int) character, which converts it into an integer value. This integer value is further converted into its respective binary value which ultimately we want using a direct toBinaryString() function which Java provides.

Now 'A' <=> 65 which when converted to binary fetches 1000001 is a 7 bit binary stream. Another value, e.g. number '6', <=> 54 when converted to binary stream fetches 110110 which is a 6 bit binary stream. When entire message stream needs to be converted into a binary stream, all these small binary streams are concatenated. At the decoder end, it would be fairly difficult for the decoder to divide the received bit stream back into ASCII values since the length of individual characters in binary are different as we just saw. To overcome this problem, all the characters are converted to 8 bit binary streams by appending zeros at the start. Hence 'A' becomes 01000001 by appending one zero and '6' is converted to 00110110 by appending two zeros at the start. The entire process of starting with a string and fetching an 8 bit binary stream is explained in the following function:

```
public int[] readFile(String passedString)

{

tempArray = new char[passedString.length()],

tempArray2 = new char[tempArray.length * 8],

intempArray = new int[tempArray.length],

longInt = new int[tempArray2.length],

String finalString = "",

for(int i=0,i<longInt.length,i++){

longInt[i] =0,

}

String str7 = "0",

String str6 = "00",

tempArray = passedString.toCharArray(),

for(int i=0,i<tempArray.length,i++) {

intempArray[i] = (int)tempArray[i],
```

```
strBinary = Integer.toBinaryString(intempArray[i]),

if(strBinary.length()==7){

strBinary = str7.concat(strBinary),

}

else if(strBinary.length()==6){

strBinary = str6.concat(strBinary),

} finalString =

finalString.concat(strBinary),

}

}
```

All the small binary sub-streams are concatenated and stored into a final large stream declared as finalString shown above. For storage purposes, the finalString is stored into a character array where each character is either 1 or 0. This entire character array is converted into an integer array (not into the ASCII integer values but just the regular integer values)

where character '1' gets replaced with integer 1 and character '0' is replaced with integer 0. Thus a large binary string of cover media is generated at the sender using the above technique. This binary string is now divided into blocks of 12 bits each. Remember that Golay codes encode 12 bit block into a 23 bit codeword using encoding mechanism. This 12 bit block is first transformed into a row matrix with twelve elements, that is, if X = 101000101110, then X is converted into a row matrix as shown in Equation 5.1:

M = [1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0] (5.1)

M is multiplied to the generator matrix G of Golay codes which have 12 rows and 23 columns. The matrix multiplication (M x G) produces a matrix A, a row matrix with 23 bits. This row matrix is one of the 4096 codewords of the Golay codes. As explained before, all these subsequent multiplications and additions are modulo 2. Hence matrix A will contain only binary numbers 1 and 0. The conversion from 12 bit block to 23 bit codeword is processed for all the subsequent binary blocks of stream created of cover media text file and all these 23 bit codewords are appended to generate a large stream of bits. This conversion and matrix multiplication is explained below in function matMultiply():

```
public int[][] matMultiply(int [][] A, int [][] B)
```

```
{
int C [][] = new int[A.length][B[0].length],
for(int i=0,i<A.length,i++) {
for(int j=0,j<B[i].length,j++) {
C[i][j] = 0,
}
}
for ( int I = 0, I < A.length, i++)
{
for (int j = 0, j < B[i].length, j++)
{
for (int k = 0, k < A[i].length, k++)
{
C[i][j] = ((C[i][j]+(A[i][k] * B[k][j]))%2),
}
}
}
return(C),
}
```

This function takes its arguments as two matrices supposed to get multiplied viz. matrix A and matrix B. The result is stored into matrix C. For matrix multiplication, the number of rows of (B) must equal the number of columns of (A). Also the resulting matrix C should have dimensions such that there are the same number of rows of (C) as (A), and the same number of columns of (C) as (B).

Hence the matrix C is declared as "int C [][] = new int[A.length][B[0].length]," where A.length specifies number of rows of A and B[0].length specifies number of columns of B. The matrix C is initialized to zero. The matrix multiplication follows the following rules: • Every element from the row of matrix A is multiplied to its corresponding

column element of the matrix B. For example, the 1st row, 3rd column element of matrix A is multiplied by the 3rd row, 1st column of matrix B and so on.

CONCLUSION

As steganography becomes more widely used in computing there are issues that need to be resolved. A wide variety of different techniques are discussed in present paper with their advantages and disadvantages. Many of currently used techniques are not robust enough to prevent detection and removal of embedded data. The use of benchmarking to evaluate techniques should become more common and more standard definition of robustness is required to help overcome this problem.

REFERENCES:

- Singh, Hitesh, Pradeep Kumar Singh, and Kriti Saroha. "A survey on text based steganography." *Proceedings of the 3rd National Conference*. Vol. 3. No. 3. Bharati Vidyapeeth's Institute of Computer Applications and Management, 2009.
- Gupta, Swati, and Deepti Gupta. "Text-steganography: Review study & comparative analysis." (2011).
- Roy, Souvik, and P. Venkateswaran. "A text based steganography technique with Indian root." *Procedia Technology* 10 (2013): 167-171.
- Thangadurai, K., and G. Sudha Devi. "An analysis of LSB based image steganography techniques." *2014 International Conference on Computer Communication and Informatics*. IEEE, 2014.
- Lockwood, Robert, and Kevin Curran. "Text based steganography." *International Journal of Information Privacy, Security and Integrity* 3.2 (2017): 134-153.
- Bhattacharyya, Souvik, Indradip Banerjee, and Goutam Sanyal. "Design and implementation of a Secure Text Based Steganography Model." *Security and Management*. 2010.
- Dudhagara, Chetan, and Kishor Atkotiya. "Experimental Study of Fractal Image Compression Algorithm." *International Journal of Computer Applications & Information Technology* 1.2 (2012).
- Dudhagara, Chetan R., and Mayur M. Patel. "A Comparative Study and Analysis of EZW and SPIHT methods for Wavelet based Image Compression." *Oriental Journal of Computer Science and Technology* 10.3 (2017): 669-673.
- Dudhagara, Chetan, and Kishor Atkotiya. "Image compression using vector quantization." (2013).